Advant® OCS

Advant Graphics

Version 1.0

User’s Guide

3BSE 017 091R101

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Automation Products AB. ABB Automation Products AB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Automation Products AB be liable for direct, indirect, special, incidental, or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Automation Products AB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Automation Products AB’s written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

TRADEMARKS

Advant is a registered trademark and AdvaSoft is a trademark of ABB Asea Brown Boveri Ltd., Switzerland.

Microsoft, Visual Basic and Windows NT are registered trademarks of Microsoft Corporation.

Copyright © ABB Automation Products AB 1999

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1 About This Book 1
    1.1.1 How to Use This Book 1
1.2 Product Overview 4
    1.2.1 What You Can Do with Advant Graphics 4
1.3 Prerequisites and Requirements 5
1.4 Related Documentation 5
1.5 Conventions 6
1.6 Terminology 7

Chapter 2 - Installation

2.1 Installation Procedures 9
2.2 Preparations 9
2.3 Installing External (Third-party) ActiveX Controls 9
    2.3.1 How to Add External ActiveX Controls to the Toolbox 9
2.4 Product Verification 10

Chapter 3 - How to Build Graphic Elements

3.1 Introduction 11
3.2 Getting Started 12
    3.2.1 User Interface 13
    3.2.2 Application Start-up 14
3.3 Working with Advant Graphics Builder 14
    3.3.1 Advant Graphics Builder Window 15
        3.3.1.1 Advant Graphics Wizard 16
        3.3.1.2 Graphic Element Browser 21
        3.3.1.3 Add Tool 21
        3.3.1.4 Add Member 21
        3.3.1.5 Expressions Window 21

CONTENTS (continued)

3.3.2 Setting Advant Options 22
  3.3.2.1 Live Data in Design mode 22
  3.3.2.2 Advanced User Mode 23
  3.3.2.3 Debug 23
3.4 Building Graphic Elements 23
  3.4.1 Starting from Advant Plant Explorer 23
  3.4.2 Starting from Advant Graphics Builder 26
3.5 Editing Graphic Elements 27
  3.5.1 Starting from Advant Plant Explorer 27
  3.5.2 Starting from Advant Graphics Builder 28
3.6 Testing Graphic Elements 29
3.7 Saving Graphic Elements 30
3.8 Deploying Graphic Elements 30
3.9 Deleting Graphic Elements 30
3.10 Working with the Toolbox 30
  3.10.1 Bounding Rectangle 30
  3.10.2 Toolbox Window 31
    3.10.2.1 General 33
    3.10.2.2 Advant Primitives 33
    3.10.2.3 Advant Subelements 33
    3.10.2.4 Advant Extended Primitives 33
  3.10.3 External ActiveX Controls 34
  3.10.4 Providing Elements from the Toolbox 35
  3.10.5 Adding Tools to the Toolbox 35
    3.10.5.1 Tools Window 36
    3.10.5.2 Tool List 37
    3.10.5.3 OK button 37
    3.10.5.4 Cancel button 37
    3.10.5.5 Apply button 37
  3.10.6 Deleting Tools from the Toolbox 37

CONTENTS (continued)

3.11 Adding Members.....37
  3.11.1 Add Member Window.....38
    3.11.1.1 Properties field.....39
    3.11.1.2 Add Member button.....39
    3.11.1.3 Delete Member button.....40
    3.11.1.4 Attribute Information fields.....40
3.12 Adding Methods/Events.....41
3.13 Adding Properties.....42
  3.13.1 Properties Window.....42
    3.13.1.1 Object Names Containing Special Characters.....44
3.14 Searching for Graphic Elements and Adding Them to a Graphic.....44
  3.14.1 Graphic Element Browser Window.....45
    3.14.1.1 Structure Selector.....46
    3.14.1.2 Object Tree View.....46
    3.14.1.3 Aspect List.....46
    3.14.1.4 Add Element button.....46
    3.14.1.5 Close button.....46
3.15 Assigning Expressions.....46
  3.15.1 Assigning Expressions to Properties.....47
  3.15.2 Expressions Window.....47
3.16 Building Expressions.....49
  3.16.1 Expression Builder Window.....51
    3.16.1.1 Edit box.....52
    3.16.1.2 Syntax buttons.....52
    3.16.1.3 Function buttons.....52
    3.16.1.4 Structure Selector.....53
    3.16.1.5 Object Tree View.....53
    3.16.1.6 Property List (with OPC items).....53
    3.16.1.7 Insert buttons.....53
3.17 Relative References.....54

CONTENTS (continued)

3.18 Copying Graphic Aspects in the Same Object Structure 56
3.19 Importing Visual Basic Projects 57

Chapter 4 - How to Configure Process Dialogs

4.1 Introduction 59
4.2 Getting Started 59
  4.2.1 User Interface 61
  4.2.2 Application Start-up 62
4.3 Working with Process Dialogs 62
  4.3.1 Process Dialog Window 62
    4.3.1.1 Main Window 63
    4.3.1.2 Object Display 66
    4.3.1.3 Configuration View 68
  4.3.2 Configuring Process Dialogs 72
4.4 Working with the Process Dialog User Profile 73
  4.4.1 Process Dialog User Profile 74
    4.4.1.1 Process Dialog Profile Definition 75
  4.4.2 Changing the Process Dialog User Profile 77

Chapter 5 - How to Build Graphic Displays

5.1 Introduction 79
5.2 Getting Started 80
  5.2.1 User Interface 80
  5.2.2 Application Start-up 81
5.3 Working with Advant Graphics Builder 82
5.4 Building Graphic Displays 83
  5.4.1 Starting from Advant Plant Explorer 83
  5.4.2 Starting from Advant Graphics Builder 84
5.5 Editing Graphic Displays 84
  5.5.1 Starting from Advant Plant Explorer 85
  5.5.2 Starting from Advant Graphics Builder 85

CONTENTS (continued)

5.6 Testing Graphic Displays 86
5.7 Saving Graphic Displays 87
5.8 Deploying Graphic Displays 87
5.9 Deleting Graphic Displays 87
5.10 Scaling Bitmap Pictures 87

Chapter 6 - How to View Graphics in Run-time

6.1 Introduction 89
6.2 Getting Started 89
6.3 Viewing Graphics 89
    6.3.1 Main Viewing Window 90
6.4 Object Marking 91
6.5 Respond to Changes in Graphics 91

Chapter 7 - Maintenance

7.1 Preventive Maintenance 93
7.2 Frequently Asked Questions (FAQ) 93

Chapter 8 - Tutorial

8.1 Creating a Graphic Element from Advant Plant Explorer 96
8.2 Building a Graphic Element 99
8.3 Changing Properties for the Graphic Element 101
8.4 Assigning Expressions to the Graphic Element 101
8.5 Building Expressions for the Graphic Element 103
8.6 Testing the Graphic Element 104
8.7 Saving a Graphic Element 105
8.8 Deploying a Graphic Element 105
8.9 Using the Graphic Element in a Graphic Display 105
8.10 Creating a Graphic Element and Using Visual Basic Code to Animate It 111

CONTENTS (continued)

Appendix A - Basic Building Blocks Provided by ABB

A.1 Introduction ... 119
A.2 Advant Primitives (Primitive Elements) ... 121
  A.2.1 AdvantArc ... 121
  A.2.2 AdvantBar ... 122
  A.2.3 AdvantBezier ... 123
  A.2.4 AdvantChord ... 124
  A.2.5 AdvantCone ... 125
  A.2.6 AdvantEllipse ... 126
  A.2.7 AdvantHscale ... 127
  A.2.8 AdvantLimit ... 128
  A.2.9 AdvantNumeric ... 129
  A.2.10 AdvantObjectTrim ... 131
  A.2.11 AdvantPie ... 133
  A.2.12 AdvantPipe ... 134
  A.2.13 AdvantPolygon ... 135
  A.2.14 AdvantPolyline ... 136
  A.2.15 AdvantRectangle ... 137
  A.2.16 AdvantText ... 138
  A.2.17 AdvantVscale ... 139
  A.2.18 Aspect Link ... 141
A.3 Primitive Element Properties ... 142
  A.3.1 Line Properties ... 143
    A.3.1.1 Line Styles ... 143
  A.3.2 Fill Properties ... 144
    A.3.2.1 Fill Styles ... 145
  A.3.3 Fill Properties with Shade ... 146
    A.3.3.1 Fill Styles ... 147

CONTENTS (continued)

A.3.4 3D Frame Properties 149
A.3.4.1 3D Effect....149
A.3.5 Smooth 150
A.3.6 Angle Properties....150
A.3.7 Point Properties 151
A.3.7.1 To Add a New Point 151
A.3.7.2 To Delete a Point 152
A.3.7.3 To Move a Point 152
A.4 Advant Subelements....153
A.4.1 AdvantBar 153
A.4.1.1 Bar Styles....155
A.4.2 AdvantValueField....158
A.4.3 AdvantObjectTrim 159
A.4.4 AdvantStatusLine....160
A.4.4.1 Status Line Styles 162

Appendix B - Expressions

B.1 Introduction....165
B.2 ABB Expressions 166
B.2.1 Object References 167
B.2.2 Syntax for Item References....168
B.2.2.1 ObjectRef....168
B.2.2.2 PropertyRef....168
B.3 Visual Basic Operators....169
B.4 If-then-else Statements....170
B.5 Paths to the OPC Server....172

INDEX

Chapter 1 Introduction

1.1 About This Book

This book is intended both for those who build, configure and use graphics (graphic elements, graphic displays and process dialogs) as well as for those who interpret and respond to changes in the graphics. It also includes information about the basic building blocks provided by ABB.

It is assumed that you are familiar with the concept of ABB Objects, Aspects, Structures etc. before you start using this book.

Information about the necessary preparations are described in *Advant Workplace Engineering Methods Reference Manual*.

This book is not the only source of instruction for Advant Graphics. ABB also offers training courses for those who use Advant Graphics.

1.1.1 How to Use This Book

The main parts of this book are:

Where to Start

If you are a first-time user, refer to Section 1.2.1, What You Can Do with Advant Graphics to learn about the basic functions and features for this software option.

Installation

Chapter 2, Installation describes some of the procedures and preparations you need to know when you load/install Advant Graphics software and external ActiveX controls.

Building Graphic Elements, Process Dialogs and Graphic Displays

Refer to Chapter 3, How to Build Graphic Elements, Chapter 4, How to Configure Process Dialogs and Chapter 5, How to Build Graphic Displays to learn what you need to do either in Advant Graphics Builder or in external applications (for example Visual Basic) to build graphic elements, process dialogs and graphic displays.

Viewing Object Details

Refer to Chapter 4, How to Configure Process Dialogs to learn more about how you use process dialogs. Chapter 6, How to View Graphics in Run-time describes how you view and respond to graphics (graphic elements and graphic displays).

Basic Building Blocks Provided by ABB

Refer to Appendix A, Basic Building Blocks Provided by ABB to learn more about the graphics provided by ABB. The Advant Primitives (primitive elements) and Advant Subelements included in this appendix are those supplied by ABB at delivery.

Assigning and Building Expressions

Refer to Section 3.15, Assigning Expressions, Section 3.16, Building Expressions and Appendix B, Expressions to learn more about expressions and how you use them.

Use of Warning, Caution, Information, and Tip

This publication includes **Warning**, **Caution**, and **Information** where appropriate to point out safety related or other important information. It also includes **Tip** to point out useful hints to the reader. The corresponding symbols should be interpreted as follows:

⚠ Warning indicates the presence of a hazard which could result in *personal injury*.

❗ Caution indicates the presence of a hazard which could result in *equipment or property damage*.

ℹ Information alerts the reader to pertinent facts and conditions.

💡 Tip indicates advice on, for example, how to design your project or how to use a certain function.

Although **Warning** hazards are related to personal injury, and **Caution** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **Warning** and **Caution** notices.

1.2 Product Overview

Advant Graphics consists of a graphic run-time viewer included in some Advant Workplace Products. It also includes the software option **Advant Graphics Builder** which is based on Microsoft Visual Basic.

For a description of the Advant Workplace Products, refer to the *Advant Workplace Products for Operators Product Guide*.

1.2.1 What You Can Do with Advant Graphics

You can do the following with Advant Graphics:

- Build graphic elements
- Configure process dialogs
- Build graphic displays
- View graphics
- Use process dialogs for information and control

You can also find the following information in this book:

- A tutorial
- Detailed information about the basic building blocks provided by ABB
- Examples of how to use expressions

1.3 Prerequisites and Requirements

Advant Graphics Builder runs on the Microsoft Windows NT operating system.

All hardware and software requirements are listed in *Advant Workplace Products for Operators Product Guide*.

It is required that you are familiar with the graphical user interface of Microsoft Windows NT. You should also a have general knowledge of common terms such as menus, buttons, drag/drop etc. It is also useful if you are familiar with Microsoft Visual Basic since Advant Graphics Builder works in conjunction with it.

1.4 Related Documentation

Table 1-1 lists all documentation related to Advant Graphics.

Table 1-1. Related Documentation

Category	Title	Description
System Administration	Advant Workplace Products Administrator's Guide	This book describes how you install the Advant Workplace Products software, including software options, and how you then perform maintenance.
Software	Advant Workplace User's Guide	This book describes everything about the user interface, for example how you present information, navigate and control objects.
	Advant Plant Explorer User's Guide	This book describes how you build hierarchically structured models of a plant or project as well as how you navigate in the structures.
Options	Advant Connectivity Library for Advant Controller 100 Series Reference Manual	This book is a set of pre-defined graphic elements for Advant Controller 100 Series.
	Advant Graphic Library for Process Industry Reference Manual	This book is a set of pre-defined graphics used by the process industry.

Table 1-1. Related Documentation (Continued)

Category	Title	Description
Other	Advant Workplace Products for Operators Product Guide	This book is a presentation of the software and software options in Advant Workplace Products for Operators. It also includes hardware and software requirements and specifications.
	Advant Workplace Engineering Methods Reference Manual	This book is intended as a guide for plant engineering using Advant Workplace Products software together with Advant Controller 100 Series.

More information about Visual Basic can be found in Microsoft's own manuals, On-line Help and On-line Documentation.

1.5 Conventions

The following conventions are used throughout this book for the presentation of material:

- The words in proper names of screen elements (for example, the title in the title bar of a window, the label for a field of a dialog box) are initially capitalized.
- Capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, press the ENTER key.
- Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the **space bar**, **comma key**, and so on.
- A plus sign is used to indicate that you must simultaneously press several keys. For example, CTRL+C indicates that you must hold down the CTRL key while pressing the C key.
- The phrase "press and release" is used to indicate that you sequentially press several keys. For example, to close a window, press and release ALT, and then F4. This indicates that you press and release the ALT key, and then you press and release the F4 function key.

- The names of push and toggle buttons are boldfaced. For example, click **OK**.
- The names of menus and menu items are boldfaced. For example, the **File** menu.
  - The following convention is used for menu operations: MenuName > MenuItem > CascadedMenuItem. For example: choose **File** > **New** > **Type**.
  - The **Start** menu name always refers to the **Start** menu on the Windows NT Task Bar.
- System prompts/messages are shown in the Courier font, and user responses/input are in the boldfaced Courier font. For example, if you enter a value out of range, the following message is displayed:

  ```
  Entered value is not valid. The value must be 0-30.
  ```

  In a tutorial, you may be told to enter the string TIC132 in a field. The string is shown as follows in the procedure:

  ```
  TIC132
  ```

1.6 Terminology

The following is a list of terms associated with Advant Graphics that you should be familiar with.

Term	Description
properties	Properties, for example color and font, belonging to currently active ActiveX Controls.
expressions	Expressions are calculations that specify subscriptions and relationship between process data and data to be displayed. The expression language is based on standard Visual Basic.

Chapter 2 Installation

2.1 Installation Procedures

The Advant Graphics software, including Advant Graphics Builder, symbol libraries, additional ActiveX libraries and other options, is installed according to the instructions in the *Advant Workplace Products Administrator’s Guide*.

2.2 Preparations

Refer to *Advant Workplace Engineering Methods Reference Manual*.

If you want to build your own animated graphics or import and filter photos, which are all useful additions to graphic elements/process displays, there are several third party products that can be used. We recommend Paint Shop Pro from JASC. Please look at http://www.jasc.com. There is a trial license for 30 days.

2.3 Installing External (Third-party) ActiveX Controls

Perform the following steps to install the “external” ActiveX Controls:

1. Follow the suppliers’ instructions for the product.

2. Add the external ActiveX Controls to the Advant Graphics Toolbox.

2.3.1 How to Add External ActiveX Controls to the Toolbox

To add external ActiveX Controls:

1. Choose **Components** in the Aspect menu. The Components window is presented.
2. Select the Control tab. A list box will appear with ActiveX Controls that can be included.
3. Click in the boxes to select one or more of the listed ActiveX Controls.

4. Click **OK**. The tools of the selected ActiveX Control are presented in the toolbox.

For license purposes you have to install these ActiveX components in all your Advant Workplaces including the server(s) according to the suppliers instructions. Graphic displays containing ActiveX components which have not been installed in an Advant Workplace will not appear on it's screen.

External ActiveX controls are NOT described in this manual or any other related document delivered from ABB.

External ActiveX controls for which licenses are needed sometimes require different licensing methods. For information about how to obtain these licenses, see the instructions from each supplier.

2.4 Product Verification

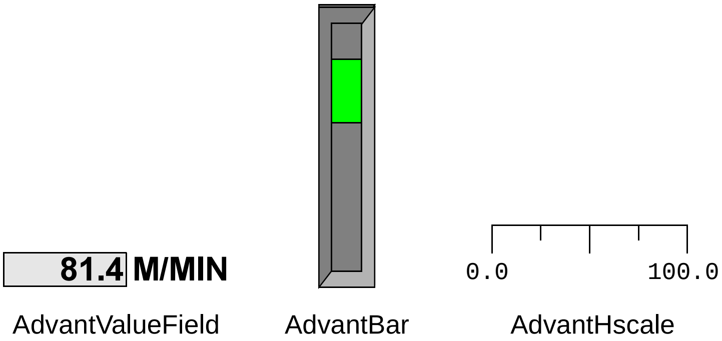
To get information regarding the installed software, choose **Help > About Advant Graphics** from the menu bar. This displays an information box with information regarding version, licensing, and so on.

To close the box, just click inside it.

Chapter 3 How to Build Graphic Elements

3.1 Introduction

A graphic element is normally part of a graphic display or process dialog and typically built up from several subelements and/or primitive elements. Some graphic elements are provided by ABB.



AdvantValueField AdvantBar AdvantHscale

Figure 3-1. Examples of Graphic Elements provided by ABB

There are no clear borderlines between the element types, but the following definitions are used in this document.

- **Primitive element**

  A simple graphic element, for example a rectangle or a polyline. Primitive elements are graphic components (ActiveX controls) that are installed into system libraries. All elements that you use which haven't been built in Advant Graphics Builder are treated as primitive elements.

- **Subelement**

  A graphic element composed of primitive elements and suitable for data type presentation.

For more information about graphic elements provided by ABB, see Appendix A, Basic Building Blocks Provided by ABB.

3.2 Getting Started

The following is a summary of the basic steps for using Advant Graphics Builder when you are working with graphic elements.

1. Familiarize yourself with Advant Graphics Builder by looking through Section 3.2.1, User Interface.
2. Either create a new graphic element (that is, Advant Graphics aspect), Section 3.4, Building Graphic Elements, or edit an existing one, Section 3.5, Editing Graphic Elements.

   If this is a new graphic element a wizard will help you to get the right settings to get started, Section 3.3.1, Advant Graphics Builder Window.
3. Advant Graphics Builder is now ready to be used. Select the tools you need from the toolbox, Section 3.10, Working with the Toolbox.
4. Build the graphic element, Section 3.4, Building Graphic Elements.
5. Add properties, Section 3.13, Adding Properties.
6. Assign expressions, Section 3.15, Assigning Expressions.

   Use the Expression Builder if you want to build more complex expressions, Section 3.16, Building Expressions.
7. Test the graphic element, Section 3.6, Testing Graphic Elements

8. Save the graphic element before you exit the application, Section 3.7, Saving Graphic Elements.
9. Deploy the graphic element, Section 3.8, Deploying Graphic Elements.

3.2.1 User Interface

Advant Graphics Builder has most of Visual Basic's features for building graphic elements, plus some specific to this system. These are:

- The Advant Graphics wizard that guides you through the steps of building graphic elements.
- The Design and Test function that enables you to build graphic elements in design mode and then check their behaviour in test mode.
- The Add Member function that allows you to add properties, methods, and events to a graphic element.
- The Expressions window that allows you to assign expressions (that is, to specify subscriptions and specify the relationship between the process data and the data that is to be displayed).
- The Add Tool function that allows you to add tools to your toolbox.
- The Graphic Element Browser in which you can search for appropriate graphic elements.
- The Deploy function which stores the graphic elements and makes them accessible for viewing. Before a graphic element is deployed it can only be seen in Advant Graphics Builder.
- The Help function that offers you access to On-line Help and On-line Documentation for Advant Graphics Builder.

These features are accessed through the pull-down menu in the menu bar or through special buttons.

3.2.2 Application Start-up

To start Advant Graphics Builder you can either create a new aspect or edit an existing one.

Create a New Aspect

1. Start Advant Plant Explorer.
2. Select a structure and an ABB Object.
3. Right-click in the aspect area and choose **New Aspect** from the context menu.
4. In the New Aspect window: Select aspect category in the structure list (Aspect System Structure/Advant Graphics/Graphic Element/Graphic Element), give the new aspect a name and click **Create**.
5. Select the new aspect in the Aspect information list, right-click and choose **Edit** from the context menu. Advant Graphics Builder and its wizard starts. (You have to follow some of the wizard's steps before you can begin building the graphic element.)

Edit an Existing Aspect

1. Select the aspect (graphic element).
2. Right-click and choose **Edit** from the context menu. Advant Graphics Builder starts with the selected graphic element.

3.3 Working with Advant Graphics Builder

Advant Graphics Builder provides several features for building and editing graphics (graphic elements/displays), and writing expressions. The Visual Basic code panes are available and the environment allows you to write normal Visual Basic code. This section gives you a description of the tools added to standard Visual Basic.

3.3.1 Advant Graphics Builder Window

Enter the Advant Graphics Builder window by going through the steps described in Section 3.2.2, Application Start-up.

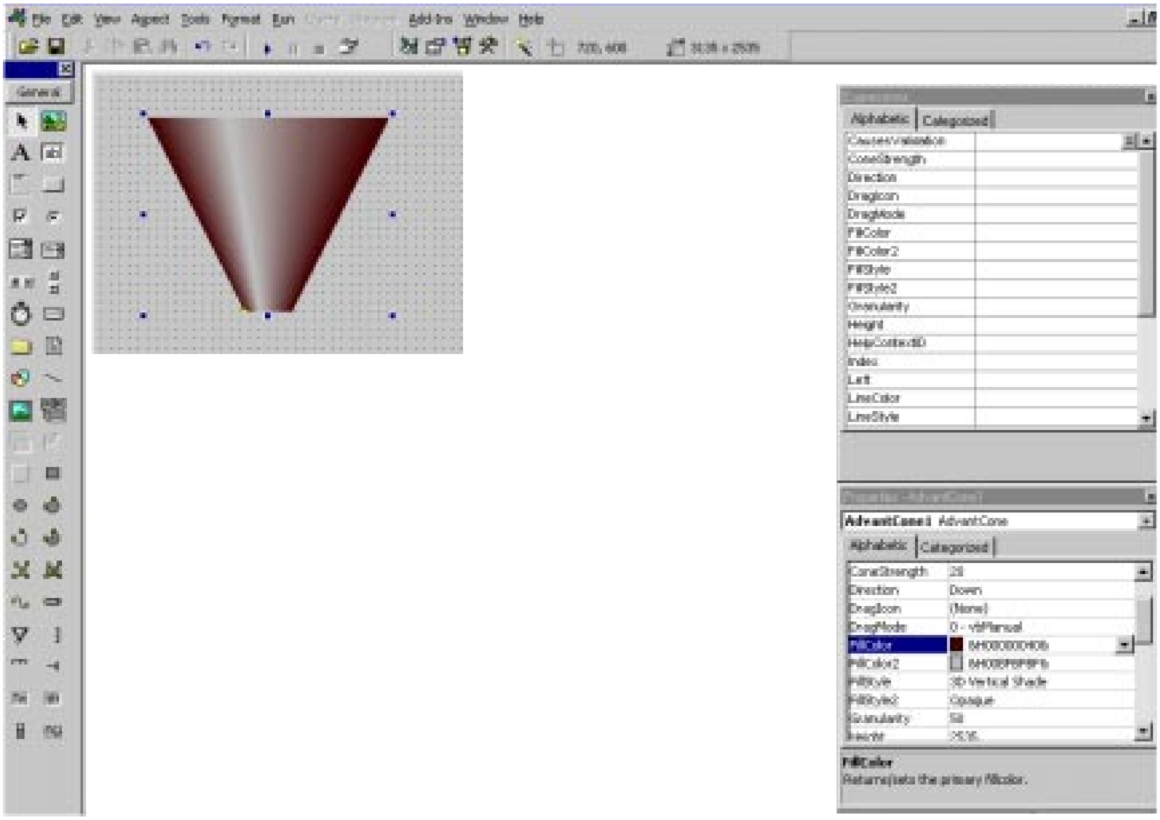


Figure 3-2. Advant Graphics Builder Window

When you are working with graphic elements, the main objectives of Advant Graphics Builder are:

- To build graphic elements. Section 3.4, Building Graphic Elements.
- To edit graphic elements. Section 3.5, Editing Graphic Elements.
- To test graphic elements. Section 3.6, Testing Graphic Elements.

- To save graphic elements. Section 3.7, Saving Graphic Elements.
- To deploy graphic elements. Section 3.8, Deploying Graphic Elements.

Several sections in this chapter **also** applies when you are working with graphic displays:

- Section 3.3.1.1, Advant Graphics Wizard
- Section 3.3.1.2, Graphic Element Browser
- Section 3.3.1.3, Add Tool
- Section 3.3.2, Setting Advant Options
- Section 3.10, Working with the Toolbox
- Section 3.14.1, Graphic Element Browser Window
- Section 3.17, Relative References
- Section 3.18, Copying Graphic Aspects in the Same Object Structure
- Section 3.19, Importing Visual Basic Projects

One section in Chapter 5, How to Build Graphic Displays, **also** applies when you are working with graphic elements:

- Section 5.10, Scaling Bitmap Pictures

3.3.1.1 Advant Graphics Wizard

The Advant Graphics wizard guides you when you start Advant Graphics Builder from Advant Plant Explorer. If you are already in Advant Graphics Builder and you want to start to build a new graphic (graphic element/display) you can either click the Wizard button in the toolbar or choose **New Aspect** in the File menu.

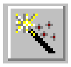
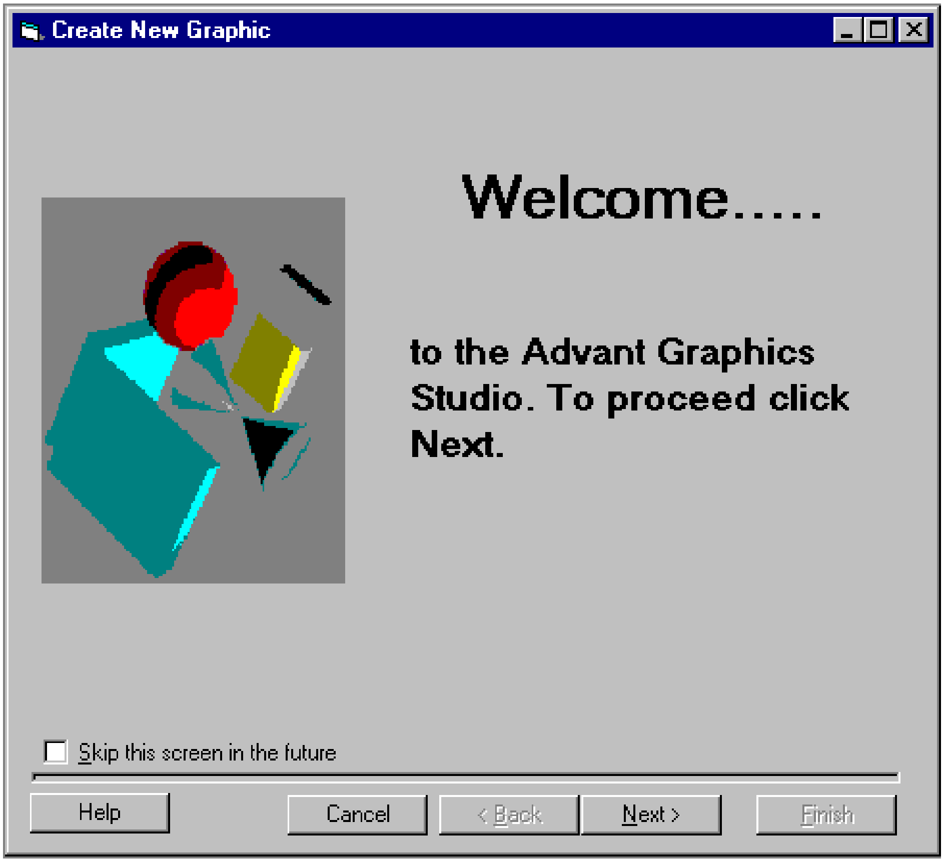


Figure 3-3. Wizard Button

The Advant Graphics wizard takes you through the procedure of creating a graphic step by step. If you click the Finish button on Page Two of the wizard, a default graphic is created. Going through the Advant Graphics wizard gives you a number of choices on how to configure the graphic. The wizard consist of four pages:

Page One

This is a welcome page. You can choose if you want to see it in the future or not by clicking the check box in the lower left corner of the page.


Create New Graphic

Welcome.....

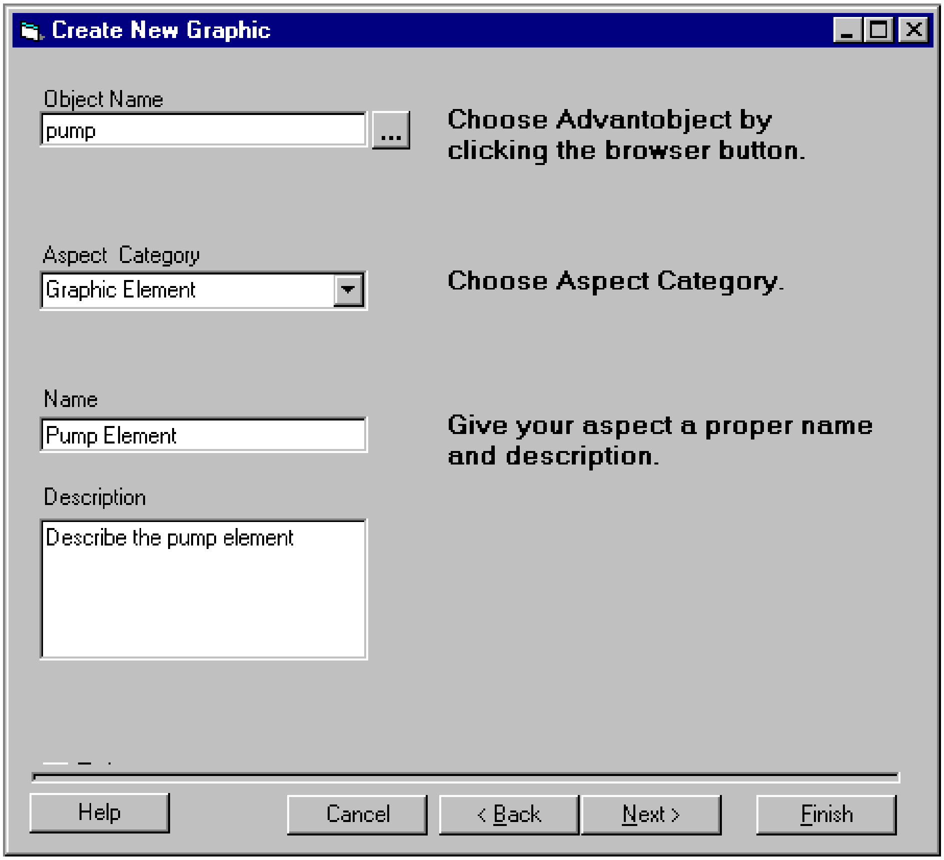
to the Advant Graphics Studio. To proceed click Next.

☐ Skip this screen in the future

Figure 3-4. Page One of the Advant Graphics wizard

Page Two

On this page you create/select an aspect. The page contains all the basic information about the aspect. If you started Advant Graphics Builder from a new graphic aspect in Advant Plant Explorer all the information in this page is already complete.



Object Name: pump

Choose Advantobject by clicking the browser button.

Aspect Category: Graphic Element

Choose Aspect Category.

Name: Pump Element

Description: Describe the pump element

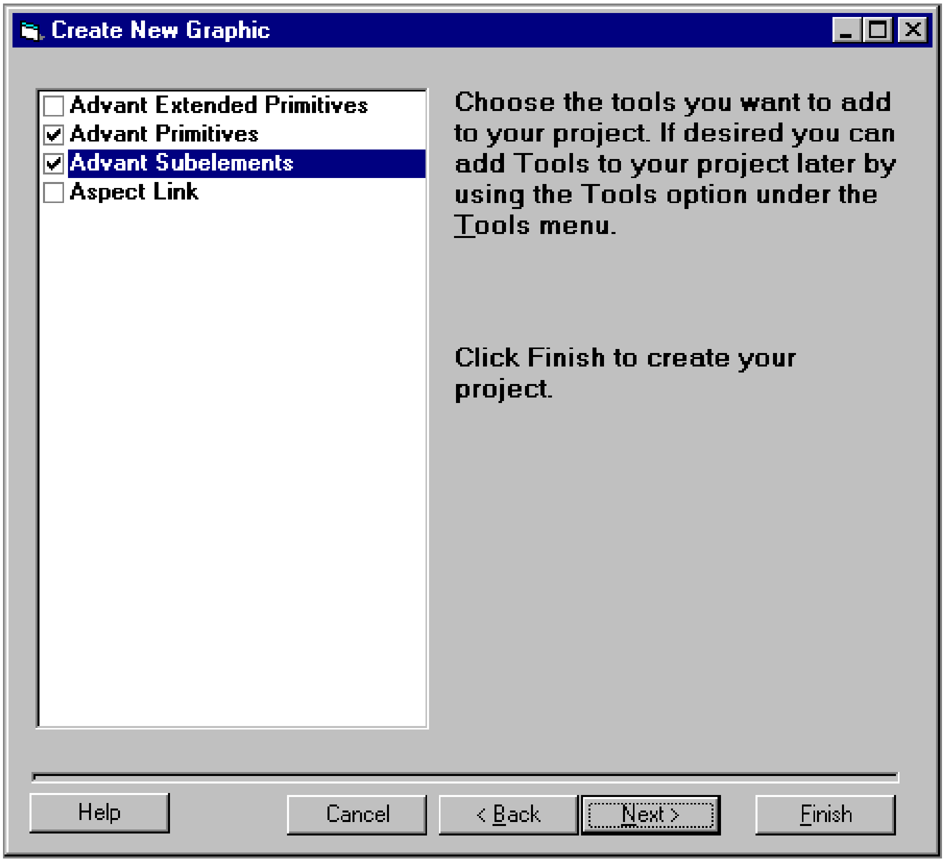
Give your aspect a proper name and description.

Figure 3-5. Page Two of the Advant Graphics wizard

Use this page to select an ABB Object name, an aspect category and a name. As an option you can add a description text.

Page Three

On this page you select the libraries of ActiveX Controls (tools) you want to use in your graphic elements and graphic displays.You automatically start on this page if you select to edit a newly created Advant Graphics aspect.



☐ Advant Extended Primitives
☑ Advant Primitives
☑ Advant Subelements
☐ Aspect Link

Choose the tools you want to add to your project. If desired you can add Tools to your project later by using the Tools option under the Tools menu.

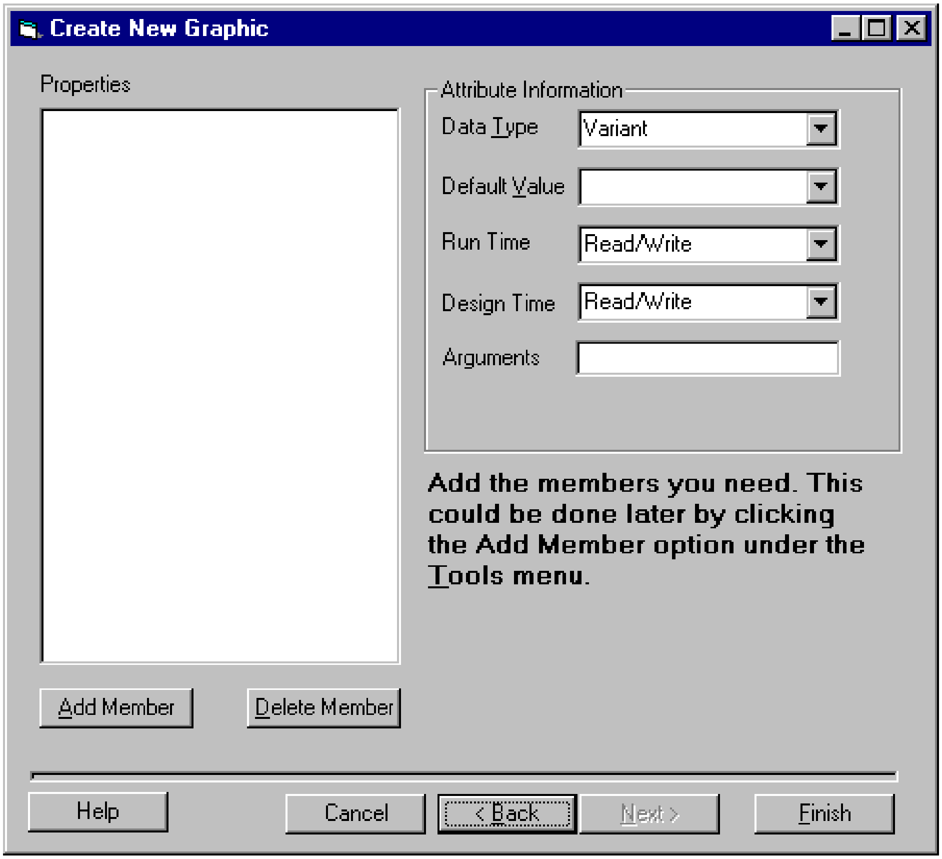
Click Finish to create your project.

Figure 3-6. Page Three of the Advant Graphics wizard

To add additional tools, see Section 3.10.5, Adding Tools to the Toolbox.

Page Four

This page is used to add members (that is, either property, method and event). See Section 3.11, Adding Members.



Data Type: Variant

Default Value:

Run Time: Read/Write

Design Time: Read/Write

Arguments:

Add the members you need. This could be done later by clicking the Add Member option under the Tools menu.

Figure 3-7. Page Four of the Advant Graphics wizard

3.3.1.2 Graphic Element Browser

With the Graphic Element Browser you can search for graphic elements that you have built before, and saved to reuse. Open the Graphic Element Browser by choosing **Element Browser** in the Tools menu in the menu bar.

3.3.1.3 Add Tool

To add tools either choose **Tools** in the Tools menu and select the appropriate check boxes in the Tools window, or select the check boxes on the third page of the Advant Graphics wizard. Either case gives you a list with check boxes where you can select the tools (that is, libraries with primitive elements, subelements and other ActiveX Controls) you need for your graphic.

3.3.1.4 Add Member

You can access the Add Member feature by choosing **Add Member** in the Tools menu or through the last page of the Advant Graphics wizard. It allows you to add properties to graphic elements.

3.3.1.5 Expressions Window

The Expressions window allows you to assign or change the expressions of the constituent components of the graphic elements. If you have closed the Expressions window, choose **Expressions Window** in the View menu to open it again.

3.3.2 Setting Advant Options

There are some optional preferences which you can set for Advant Graphics Builder. You can decide to use live data in Design mode, to have additional Visual Basic functionality and to have access to a debug menu.

Open the Advant options window by choosing **Advant Options** in the Tools menu.

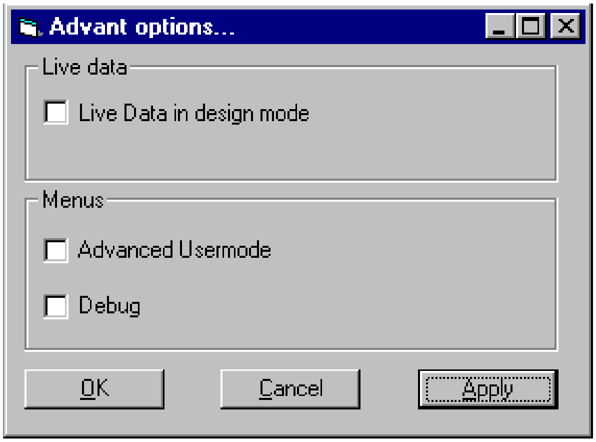


Figure 3-8. Advant options window

3.3.2.1 Live Data in Design mode

The Live Data option enables you to display dynamic data in graphic elements when you are using graphic elements in Design mode. This is useful if you don't want to switch to Test mode when you are using graphic elements.

Note that the graphics (graphic elements/displays) you are working with will only show dynamic behaviour in Test mode (with the above exception).

Also note that live data results in a higher CPU usage, which in turn means slower opening times for aspects.

Click in the check box to select Live data in Design mode.

3.3.2.2 Advanced User Mode

The Advanced User Mode option gives you access to more menus and functions in Visual Basic. If you haven't selected Advanced User Mode some menus or menu items are dimmed, which means that they are unavailable.

Click in the check box to select Advanced User Mode.

3.3.2.3 Debug

The Debug option gives you access to an additional debug menu.

Click in the check box to select Debug.

3.4 Building Graphic Elements

There are two ways in which you can start building graphic elements:

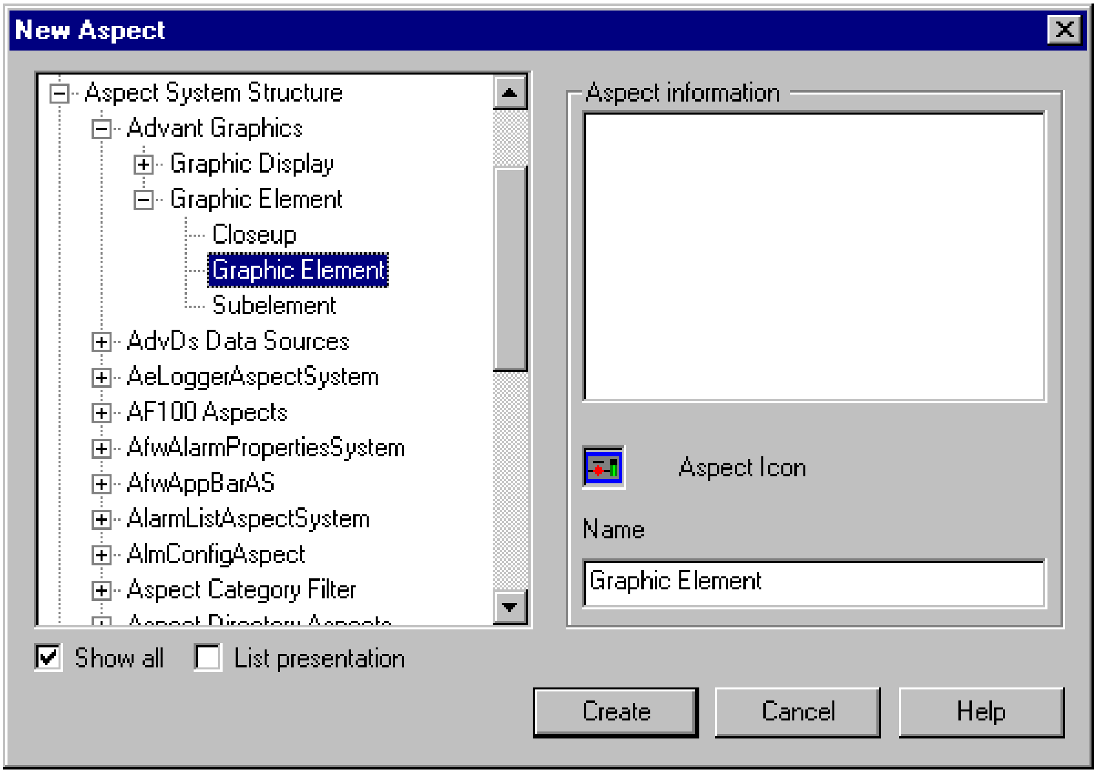
- Starting from Advant Plant Explorer
- Starting from Advant Graphics Builder

3.4.1 Starting from Advant Plant Explorer

Below follows a description of how to build a graphic element when you start from Advant Plant Explorer:

1. Start Advant Plant Explorer.
2. Select a structure.
3. Select an ABB Object.
4. Add an aspect by right-clicking in the aspect area and choose **New Aspect** from the context menu. The New Aspect window is presented.

5. Select the aspect category Graphic Element in the structure list.



Name: Graphic Element

☑ Show all ☐ List presentation

Figure 3-9. New Aspect Window

6. Give the new aspect a name and click **Create**. The selected aspect is added to the Aspect information list.
7. Right-click it and choose **Edit** from the context menu. This starts the Advant Graphics wizard in Advant Graphics Builder. Page Three in the wizard is presented.
8. Click **Next**. Page Four is presented. This page is used to add members. If you prefer you can always do it later. See Section 3.11, Adding Members.

9. Click **Finish**. It is now possible to start building a graphic element.

   Notice that an element workspace covers a much smaller area than a display workspace and that they use different background colors (grey and green).

10. Click the icon for the desired tool (ActiveX control) in the toolbox. The cursor changes to a crosshair.
11. Place the cursor at any point in Advant Graphics Builder's workspace.
12. Click and hold the left mouse button.
13. Drag the cursor diagonally to the opposite corner of the bounding rectangle.
14. Release the mouse button. The selected control now appears inside a bounding rectangle in the workspace.

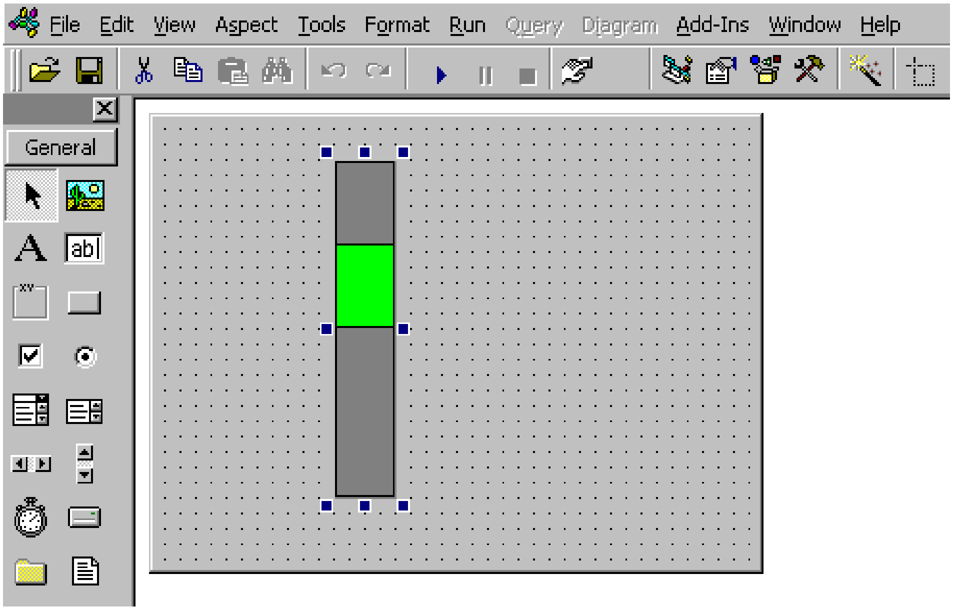


Figure 3-10. A Graphic Element in the Advant Graphics Builder Window

15. Continue building the graphic element until it is complete.

Special for Copy/Paste in this release of Advant Graphics Builder (Release 1): Expressions for a control are not included when you copy it. You must enter the same expressions for the copy.

3.4.2 Starting from Advant Graphics Builder

Below follows a description of how to build a graphic element when you start from Advant Graphics Builder.

1. Either click the Wizard button in the toolbar or choose **New Aspect** in the File menu. This starts the Advant Graphics wizard in Advant Graphics Builder. Page One in the wizard is presented.
2. Click **Next**. Page Two in the wizard is presented.
3. Click on the browser button for the Object Name field and select an ABB Object.
4. Click on the Object Name field's Browser button. The Object Browser window is presented.
5. Select the ABB Object name and click **OK**.
6. Then you select the aspect category Graphic Element from the combo box.
7. Give your graphic element a name.
8. Enter a description text (optional).
9. Click **Next**. Page Three in the wizard is presented.
10. Click **Next**. Page Four is presented. This page is used to add members. If you prefer you can always do it later. See Section 3.11, Adding Members.
11. Click **Finish**. It is now possible to start building a graphic element.

12. Click the icon for the desired tool (ActiveX control) in the toolbox. The cursor changes to a crosshair.
13. Place the cursor at any point in Advant Graphics Builder's workspace.
14. Click and hold the left mouse button.
15. Drag the cursor diagonally.
16. Release the mouse button. The selected graphic element now appears inside a bounding rectangle in the workspace.
17. Continue building the graphic element until it is complete.

Special for Copy/Paste in this release of Advant Graphics Builder (Release 1): Expressions for a control are not included when you copy it. You must enter the same expressions for the copy.

3.5 Editing Graphic Elements

There are two ways in which you can start editing graphic elements:

- Starting from Advant Plant Explorer
- Starting from Advant Graphics Builder

3.5.1 Starting from Advant Plant Explorer

Below follows a description of how to edit a graphic element when you start from Advant Plant Explorer.

1. Select the ABB Object and the graphic aspect (that is, the graphic element) which you want to edit.
2. Right-click on the aspect in the aspect area. A context menu appears.

3. Choose **Edit** from the menu. Advant Graphics Builder now starts with the selected graphic element.
4. Edit the graphic element.
5. Save your changes.

3.5.2 Starting from Advant Graphics Builder

Below follows a description of how to edit a graphic element when you start from Advant Graphics Builder.

1. Either click the Open button in the toolbar or choose **Open Aspect** in the File menu. The Open Aspect window is presented.
2. Select the ABB Object and the graphic aspect (that is, the graphic element) which you want to edit.
3. Either click **Open** or double-click on the selected aspect.

Another way, which is even faster, is to use the drag-and-drop function. Simply drag the selected aspect to the white area of Advant Graphics Builder’s workspace.

4. Edit the graphic element.
5. Save your changes.

3.6 Testing Graphic Elements

Graphic elements are built in Design mode. However, the data subscriptions are not set up, expressions (see Section 3.15, Assigning Expressions) are not executed, controls are not animated, etc. To check this you use the Test mode.

The Test mode provides functionality to:

- Check behavior of a graphic element.
- Check that all object references are resolved.

Hence, when you decide to switch from Design mode to Test mode your graphic element will set up a subscription, and values, animation, etc. are continuously updated.

To test a graphic element:

1. Switch to test mode by clicking the Test Mode button on the toolbar, or choose **Test Mode** under the Run menu.
2. Check the behavior of the graphic element.
3. Click the Design Mode button on the toolbar, or choose **Design Mode** under the Run menu to return to design mode.

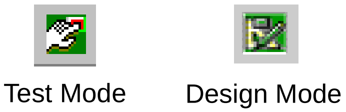


Figure 3-11. Test Mode and Design Mode Buttons

Using AF100 Test OPC Server (the off-line data source)

Advant Graphics contains a way to test your graphical elements, process dialogs and process graphics by using the off-line data source, the AF100 Test OPC Server.

See the *Advant Workplace Products Administrators Guide*, chapter 3 for information on how to connect the OPC server and how to use the Uploader aspect.

3.7 Saving Graphic Elements

To save a graphic element you either click the Save button in the toolbar or choose **Save** from the File menu. You do not have to save the graphic in any particular location. The system decides where to store all graphics.

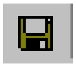


Figure 3-12. Save Button

3.8 Deploying Graphic Elements

To deploy a graphic element you choose **Deploy** from the File menu. The Deploy function is used to make it possible to view graphic element in the Advant Workplace Window. If a graphic element is not deployed you can only see it in Advant Graphics Builder.

3.9 Deleting Graphic Elements

To delete a graphic element, right-click it in Advant Plant Explorer and choose **Delete** from the context menu.

3.10 Working with the Toolbox

The toolbox provides the chosen primitive elements and subelements, used to build graphics.

3.10.1 Bounding Rectangle

All elements are drawn inside a bounding rectangle. The position properties Height, Width, Left and Top correspond to the bounding rectangle.

When the element is selected the bounding rectangle has handles on the corners and sides.

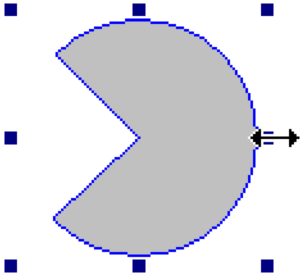


Figure 3-13. Bounding Rectangle

To change the size of the bounding rectangle, either drag the handles or enter the desired values in the Height and Width properties (the element is fitted to the bounding rectangle).

To move the element place the cursor inside the bounding rectangle and drag while holding down the left mouse button, or enter the desired values in the Left and Top properties.

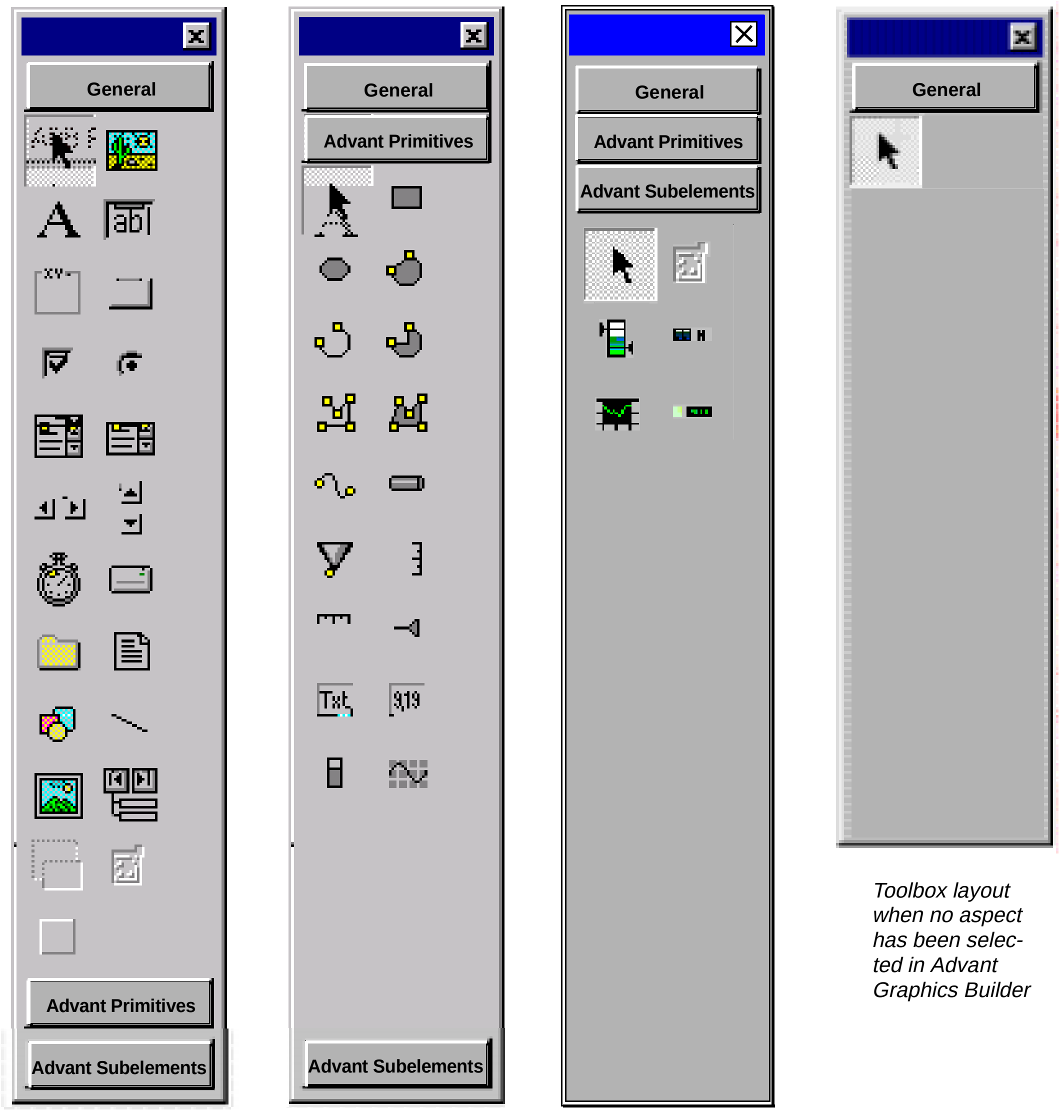
For information about how you use the properties window, see Section 3.13.1, Properties Window.

3.10.2 Toolbox Window

Open the toolbox by clicking on the Toolbox button in the toolbar or choose **View Toolbox** in the View menu.

The main objectives of the toolbox window are:

- Provide primitive elements and subelements. Section 3.10.4, Providing Elements from the Toolbox.
- Provide External ActiveX Controls. Section 3.10.4, Providing Elements from the Toolbox.



Visual Basic icons under the **General** button

ABB icons under the **Advant Primitives** button

ABB icons under the **Advant Subelements** button

Toolbox layout when no aspect has been selected in Advant Graphics Builder

Figure 3-14. Examples of different layouts of the Toolbox

The available basic toolbox buttons are:

3.10.2.1 General

This is Visual Basic's own set of ActiveX Controls (tools). They are always included in the toolbox. Click on the **General** button to display them.

3.10.2.2 Advant Primitives

These are primitive elements provided by ABB. Click on the button **Advant Primitives** in the toolbox to display the set of available primitive elements. Also included is the special element Aspect Link.

If you don't select Advant Primitives and Aspect Link on Page Three in the wizard, you won't have access to this library of ActiveX Controls (tools).

The primitive elements provided by ABB (including Aspect Link) are described in Section A.2, Advant Primitives (Primitive Elements).

3.10.2.3 Advant Subelements

These are subelements. Click on the button **Advant Subelements** in the toolbox to display the set of available subelement.

If you don't select Advant Subelements on Page Three in the wizard, you won't have access to this library of ActiveX Controls (tools).

The subelements provided by ABB are described in Section A.4, Advant Subelements.

3.10.2.4 Advant Extended Primitives

These are additional primitive elements provided by ABB. Click on the button **Advant Extended Primitives** in the toolbox to display the set of available primitive element.

If you don't select Advant Extended Primitives on Page Three in the wizard, you won't have access to this library of ActiveX Controls (tools).

The additional primitive elements provided by ABB are described in *Advant Graphic Library for Process Industry Reference Manual*.

3.10.3 External ActiveX Controls

ActiveX controls not sold by ABB are not installed as part of the product. You have to install them yourself (see Section 3.10.5, Adding Tools to the Toolbox).

For license purposes you have to install these ActiveX components in all your Advant Workplaces. Graphic displays containing ActiveX components which have not been installed in a Advant Workplace, will not be appear on it's screen.

External ActiveX Controls are NOT described in this manual or any other related document delivered from ABB.

How to Add External ActiveX Controls

To add external ActiveX Controls:

1. Choose **Components** in the Aspect menu. The Components window is presented.
2. Select the Control tab.
3. Click in the boxes to select one or more of the listed ActiveX Controls.
4. Click **OK**. The tools of the selected ActiveX Control are presented in the toolbox.

When you create a new graphic element it is automatically added to the list in the Components window and remains there even if it is deleted in Advant Plant Explorer. Don't try to fix a graphic display by creating a new graphic element with the same name as the one you have deleted. It won't work. Instead, create a new graphic element and let it replace the old (deleted) graphic element in the graphic display.

3.10.4 Providing Elements from the Toolbox

You choose the element, that is the primitive element or subelement, you want from the toolbox. When you click on the Tool button the cursor changes to a crosshair. Move the cursor where you want to place the element in the workspace. Left-click and move the cursor diagonally to a size of your choice. The element appears in a bounding rectangle. The element is fitted to the size of the rectangle. The size and position of the bounding rectangle can be changed as described in Section 3.10.1, Bounding Rectangle.

Primitive elements and subelements provided by ABB are described in Appendix A, Basic Building Blocks Provided by ABB.

Some additional primitive elements provided by ABB are described in *Advant Graphic Library for Process Industry Reference Manual*.

External ActiveX Controls are NOT described in this manual or any other related document delivered from ABB.

3.10.5 Adding Tools to the Toolbox

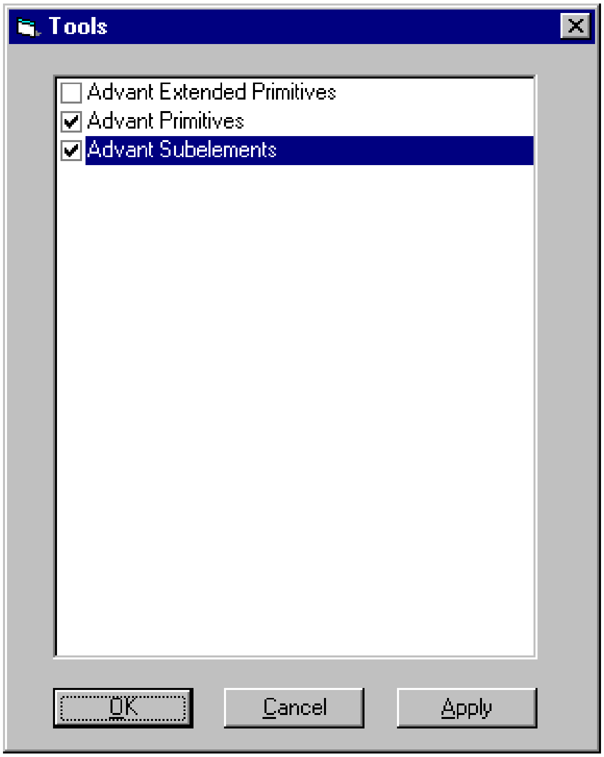
The Add Tools function allows you to add tools to your toolbox.

To add tools:

1. Choose **Tools** in the Tools menu. The Tools window is presented.
2. Click in the boxes to select one or more of the listed tools.
3. Click **OK**. The selected tools are presented in the toolbox.

3.10.5.1 Tools Window

Open the Tools window by choosing **Tools** in the Tools menu. If you are creating a new graphic, you can find the same feature on Page Three of the Advant Graphics wizard. Section 3.4, Building Graphic Elements describes how you reach the Advant Graphics wizard.



☐ Advant Extended Primitives
☑ Advant Primitives
☑ Advant Subelements

Figure 3-15. Tools Window

The sole objective of the Tools window is to add tools to your toolbox. Section 3.10.5, Adding Tools to the Toolbox.

The fields and buttons of the Tools window are:

3.10.5.2 Tool List

A list of all tools installed into your Advant Workplace.

3.10.5.3 OK button

Click **OK** to add the selected tool(s) to the toolbox and to close the Tools window.

3.10.5.4 Cancel button

Click **Cancel** to quit the Tools window with no changes to the toolbox.

3.10.5.5 Apply button

Click **Apply** to add the tool(s) you have selected.

3.10.6 Deleting Tools from the Toolbox

Deleting tools from the toolbox is described in Visual Basic's On-line Help file.

3.11 Adding Members

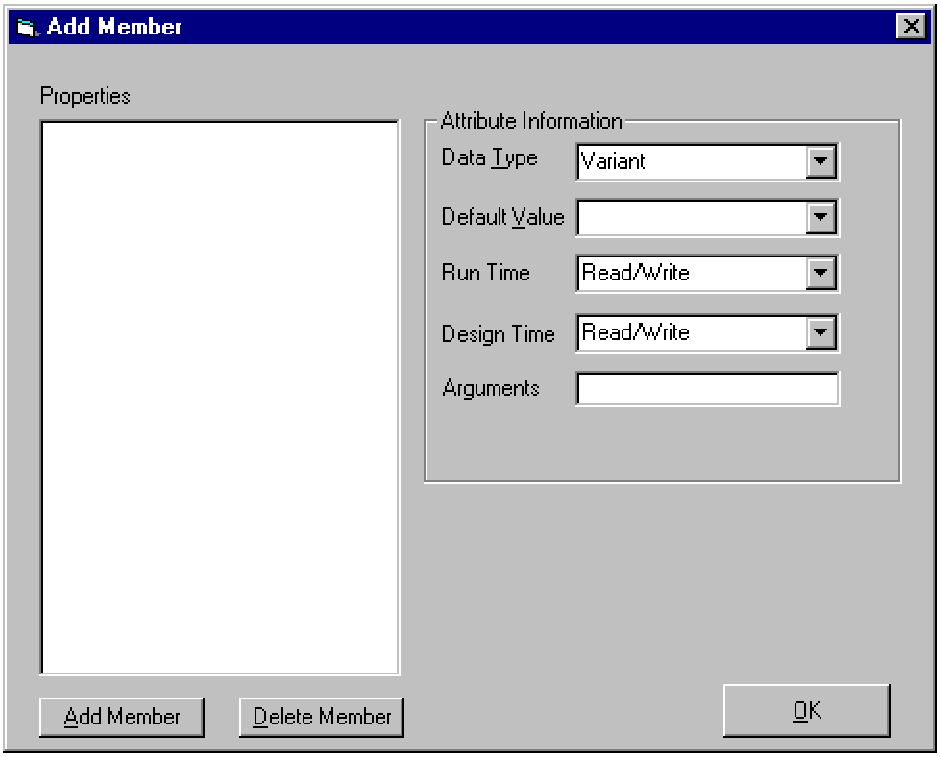
The Add Member feature is accessed through the Tools menu in the menu bar or the Wizard button in the toolbar. If you are creating a new graphic, you can find the same feature on Page Four of the Advant Graphics wizard. It allows you to add members to graphics. Member is a collective name for property, method, and event. The member is added to the entire graphic and not to any of its single elements.

You can find more details about adding properties, methods and events in the Visual Basic's On-line Help file.

3.11.1 Add Member Window

Open the Add Member window by choosing **Add Member** in the Tools menu. If you are creating a new graphic, you can find the same feature on Page Four of the Advant Graphics wizard.

The Add Member window is the fourth and last page of the Advant Graphics wizard. Section 3.4, Building Graphic Elements describes how you reach the Advant Graphics wizard.



Data Type: Variant
Run Time: Read/Write
Design Time: Read/Write

Figure 3-16. Add Member Window

The main objectives of Add Member window are:

- Add methods and events. Section 3.12, Adding Methods/Events.
- Add properties. Section 3.13, Adding Properties.

The buttons and fields of the Add Member window are:

3.11.1.1 Properties field

A list of all members that have been added to the graphic element. You can only see the member(s) you are currently working with.

3.11.1.2 Add Member button

Press **Add Member** to present the Add Member dialog box, where you enter a name and select the type (Property, Method or Event).

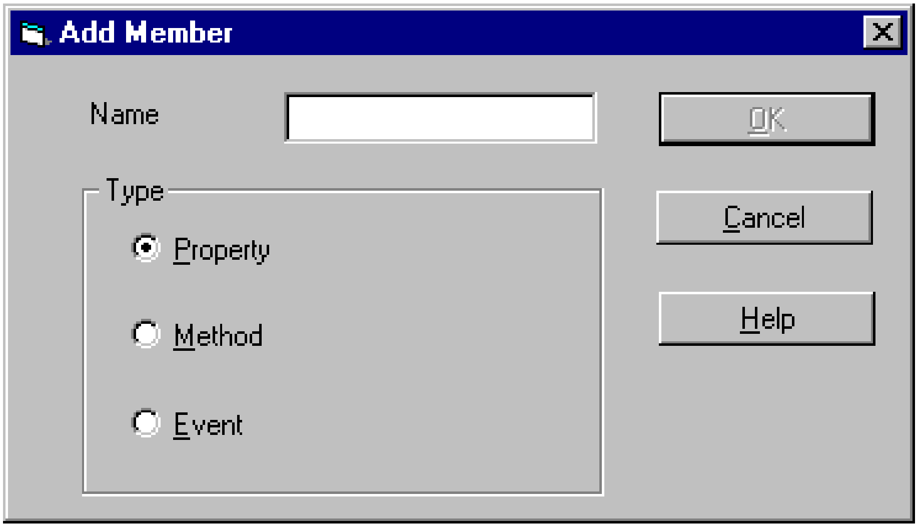


Figure 3-17. Add Member Dialog box

Don't try to add two or more members with the same name. Doing so will cause a deploy failure.

Member Types

Property is a named attribute of an object. Properties define object characteristics, such as size and name, or state of an object, such as enabled or disabled. Properties are visible in the Expressions and Properties windows.

Method is a built-in routine attached directly to an ActiveX control. It can be called from a user of graphic elements.

Event is an action, recognized by an object, for which a user of graphic elements can write code to respond to you. Events can be generated by a user action (such as clicking the mouse or pressing a key), by program code, or by the system, as with timers.

3.11.1.3 Delete Member button

Select a member and click **Delete Member** to remove it from the list.

3.11.1.4 Attribute Information fields

Attribute Information fields in the Add Members window: Data Type, Default Value, Run Time, Design Time and Arguments.

Data Type

Select one of the available data types (AdvantObject, Boolean, Color, Date, Double, Font, Integer, Long, Picture, Single, String or Variant) in this field.

You can find more details about the different data types in the Visual Basic's On-line Help file.

Default Value

Enter a default value in this field. The value depends on the selected data type.

Run Time

Select one of the available alternatives (Read, Read/Write, Write) which decides how the property will act at run time.

Design Time

Select one of the available alternatives (Read, Read/Write, Write) which decides how the property will act at design time.

Arguments

This information field is only available when you select Method as the member type. Enter an argument in this field. The argument must follow the Visual Basic syntax.

3.12 Adding Methods/Events

To add methods and events you follow the steps below.

1. Click the Add Member button in the Add Member window, or on the third page of the Advant Graphics wizard.
2. Give the member an appropriate name.
3. Select type: Method or Event. You can only do one at a time.
4. Select a data type and possibly enter an argument.
5. Click **OK**. The member is now added to the member list. You can only see the member(s) you are currently working with. Other members, that may have been added earlier, you can find in the Visual Basic's code.

3.13 Adding Properties

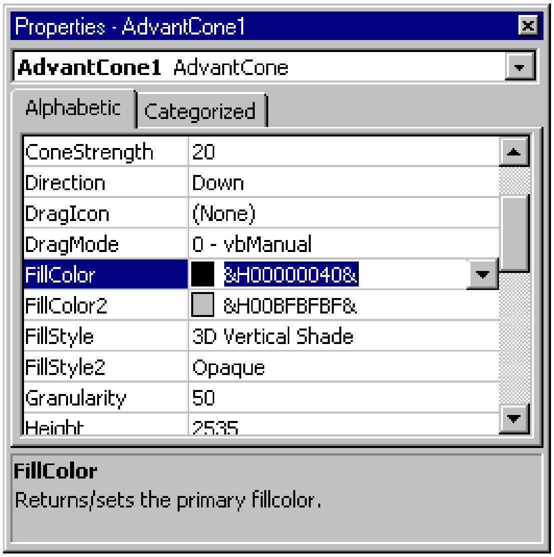
To add properties you follow the steps below.

1. Click the Add Member button in the toolbar, or on the third page of the Advant Graphics wizard.
2. Give the member an appropriate name.
3. Select type: Property.
4. Enter/select data type, default value, runtime behavior, design time behavior and attributes in the Attributes Information fields.
5. Click **OK**. The member is now added to the list in the Properties field. You can only see the member(s) you are currently working with. Other members, that may have been added earlier, you can find in the Visual Basic's code.

3.13.1 Properties Window

The Properties window is one of the standard windows in Visual Basic. It shows all the properties belonging to the currently active ActiveX Control. The properties can either be sorted in alphabetic order (with the Alphabetic tab) or in categories (with the Categorized tab). The categories are presented as a tree structure. By default, all categories are expanded. The available categories depend on the categories exposed by the control.

Open the Properties window by choosing **Properties Window** in the View menu.



ConeStrength	20
Direction	Down
DragIcon	(None)
DragMode	0 - vbManual
FillColor	■ &H00000040&
FillColor2	□ &H00BFBFBF&
FillStyle	3D Vertical Shade
FillStyle2	Opaque
Granularity	50
Height	2535

FillColor
Returns/sets the primary fillcolor.

Figure 3-18. Properties Window (with properties in alphabetic order)

Choosing colors and fonts for a graphic element is done through color and font dialog boxes which you open by clicking on the "down-arrow" buttons at the far right of the Properties window.

When exporting a graphic element to another system, the color or font relies on the target system having the corresponding color and font definitions.

A value entered in the Properties window is ignored at runtime/test mode if an expression has been entered for the same property in the Expressions window. The value is ignored as long as there is an expression in the Expressions window. All properties in the Properties window can, with the exception of Custom and Name, be found in the Expressions window.

You can find more details about the Properties window in the Visual Basic’s On-line Help file.

3.13.1.1 Object Names Containing Special Characters

Currently object names are UNICODE strings that can contain all kinds of characters except those with a special meaning (/ . + = - ++ [ ]).

Avoid using those special characters. If, on the other hand, those characters have been used for object names which you want to search for, or enter in the Properties window, add a preceding escape-character in the form of a back-slash (\).

3.14 Searching for Graphic Elements and Adding Them to a Graphic

To search for graphic elements and adding them to a graphic follow the steps below.

1. Open the Graphic Element Browser by selecting **Element Browser** in the Tools menu.
2. Select a structure in the Structure selector to the left.

   All ABB Objects belonging to the selected structure are presented in the Object Tree View Area.
3. Select an ABB Object.

   All aspects associated with that object are presented to the right in the Aspect List.
4. Select the aspect (graphic element) you want to add to the graphic.
5. Press the **Add Element** button.

💡 You can also use drag-and-drop to add a graphic element from any aspect list (for example in Advant Plant Explorer or the Graphic Element Browser).

3.14.1 Graphic Element Browser Window

Open the Graphic Element Browser by choosing **Element Browser** in the Tools menu.

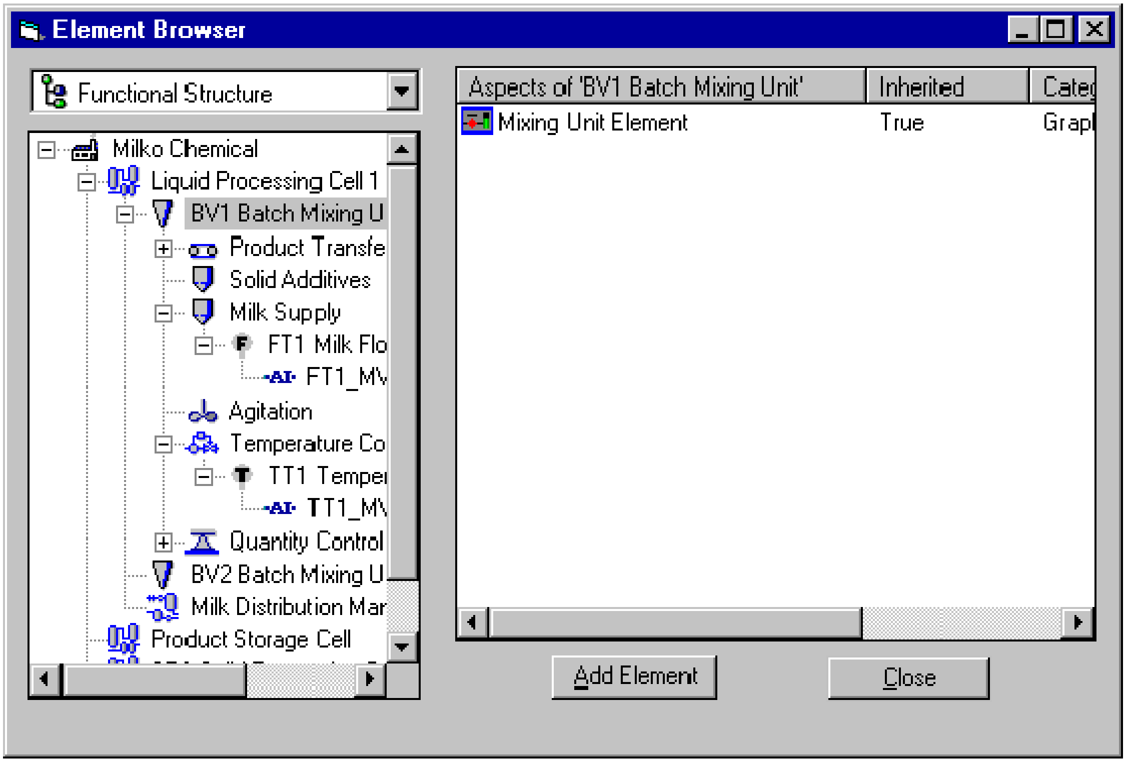


Figure 3-19. Graphic Element Browser Window

The main objectives of the Graphics Element Browser window are:

- Search for graphic elements that you have built before, and saved to reuse. Section 3.14, Searching for Graphic Elements and Adding Them to a Graphic.
- Select and add graphic elements to the graphic. Section 3.14, Searching for Graphic Elements and Adding Them to a Graphic.

The Graphics Element Browser specific fields and buttons are:

3.14.1.1 Structure Selector

The structure selector is a drop-down list from which you can select the structure to browse.

3.14.1.2 Object Tree View

The area where the selected structure is presented.

3.14.1.3 Aspect List

This list shows the graphic element aspects associated with a selected Object.

3.14.1.4 Add Element button

Click on this button when you want to add a selected graphic element to your graphic.

3.14.1.5 Close button

Click on this button when you want to close the window.

3.15 Assigning Expressions

Almost every property in graphic elements can be assigned expressions. These expressions can be set or changed via the Expressions window. The window can be opened from the View menu.

You can write expressions in two different ways.

- Use the Expression Builder. This is the preferred tool, since it gives support in how to enter the expressions.
- Write directly in the Expressions window.

Expressions are used for performing two tasks:

- Specify subscriptions.
- Specify a relationship between process data and data to be displayed.

The expression language is based on standard Visual Basic. However, some additions are made to easily set up a subscription.

3.15.1 Assigning Expressions to Properties

All properties can be assigned an expression. To assign an expression to a property:

1. Select a graphic element.
2. Select **Expressions Window** in the View menu. The Expressions window is presented.
3. From the list of the properties in the left column, select the name of the property you want to assign an expression.
4. Assign an expression to the property, by either:
   - Clicking in the property's Edit area (that is, in the right column of the Expressions window) and entering the expression you want to add, or
   - Clicking on the "equal" button (=) to display the Expression Builder and in it enter the expression.

For more information about the writing syntax for expressions, see Appendix B, Expressions.

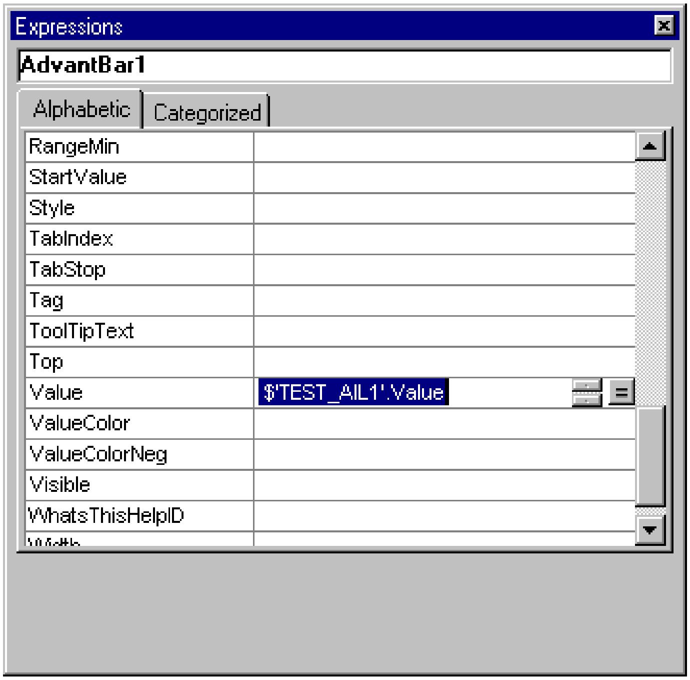
To assign an expression to a property directly in the Expressions window:

3.15.2 Expressions Window

The Expressions window shows all the properties, and their expressions, belonging to the currently active control. The properties can either be sorted in alphabetic order (with the Alphabetic tab) or in categories (with the Categorized tab). The categories are presented as a tree structure. By default, all categories are expanded. The available categories depend on the categories exposed by the control.

At the far right, the Expressions window contains a button with an equal sign (=). This button appears when you click in an edit area (that is, the right column). Clicking this button brings up the Expression Builder.

Open the Expressions window by selecting **Expressions Window** in the View menu.



RangeMin	
StartValue	
Style	
TabIndex	
TabStop	
Tag	
ToolTipText	
Top	
Value	$'TEST_AIL1'.Value
ValueColor	
ValueColorNeg	
Visible	
WhatsThisHelpID	

Figure 3-20. Expressions Window (with properties in alphabetic order)

An expression entered in the Expressions window overrides a value for the same property in the Properties window. All properties in the Properties window can, with the exception of Custom and Name, be found in the Expressions window.

Avoid using special characters for the properties (Section 3.13.1.1, Properties Window).

Special for Copy/Paste in this release of Advant Graphics Builder (Release 1): Expressions in a control are not included when you copy it. You must enter the same expressions in the copy.

The main objectives of the Expressions window are:

- Provide access to the Expression Builder. Section 3.16.1, Expression Builder Window.
- Assigning Expressions Section 3.15, Assigning Expressions.

3.16 Building Expressions

All properties can be assigned an expression. Expressions can either be written directly in the Expressions window or generated by the Expression Builder.

Building Expressions directly in the Expressions window

To build and assign an expression to a property directly in the Expressions window:

1. Select a graphic element.
2. Select **Expressions Window** in the View menu. The Expressions window is presented.
3. From the list of the properties in the left column, select the name of the property you want to assign an expression.
4. Click in the property's edit area (that is, in the right column), enter the expression you want to add.
5. Close the Expressions window.

Building Expressions in the Expression Builder window

To build and assign an expression to a property in the Expression Builder window:

1. Select a graphic element.

2. Select **Expressions Window** in the View menu. The Expressions window is presented.
3. From the list of the properties in the left column, select the name of the property you want to assign an expression.
4. Click the “equal” button (=) at the far right in the property’s edit area in the Expressions window. The Expression Builder window is presented.

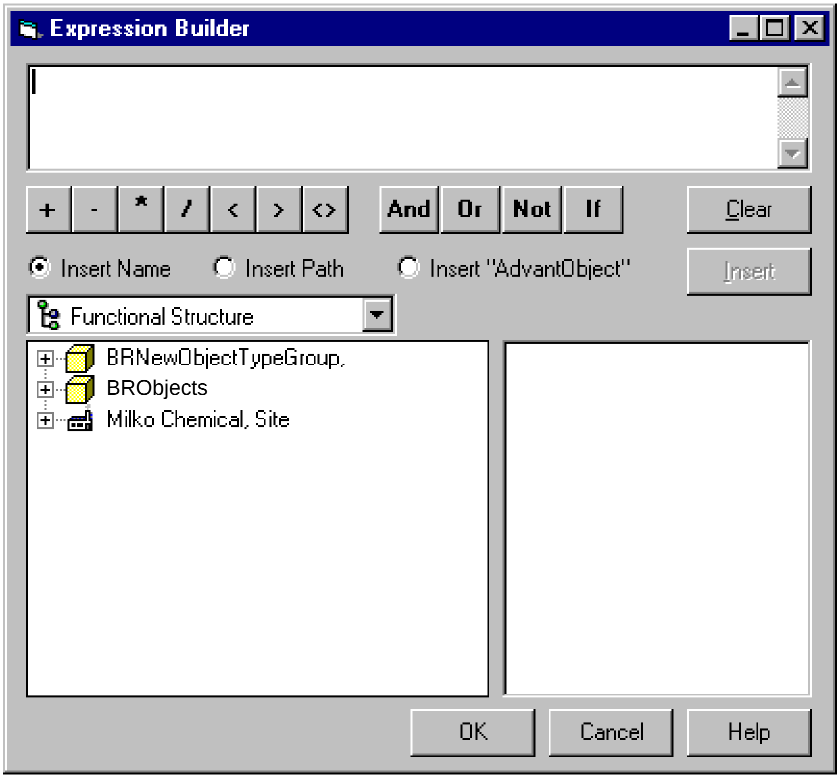


Figure 3-21. Expression Builder window

5. Build any expression you want to add by entering it in the edit box and clicking **OK**.

3.16.1 Expression Builder Window

You can either write everything directly in the edit box, or insert names of ABB Objects and OPC items from the Object Tree Area and the Property List, as well as add symbols and names with syntax buttons (+, -, /, *, <, >, <>, And, Or, Not, If).

Verify that the data type of the operand is boolean when you use the Not operator in Visual Basic. Otherwise, the Not operator will not work as expected.

For more information about the writing syntax for expressions, see Appendix B, Expressions.

Open the Expressions Builder window by clicking the “equal” button (=) at the far right in an edit area in the Expressions window.

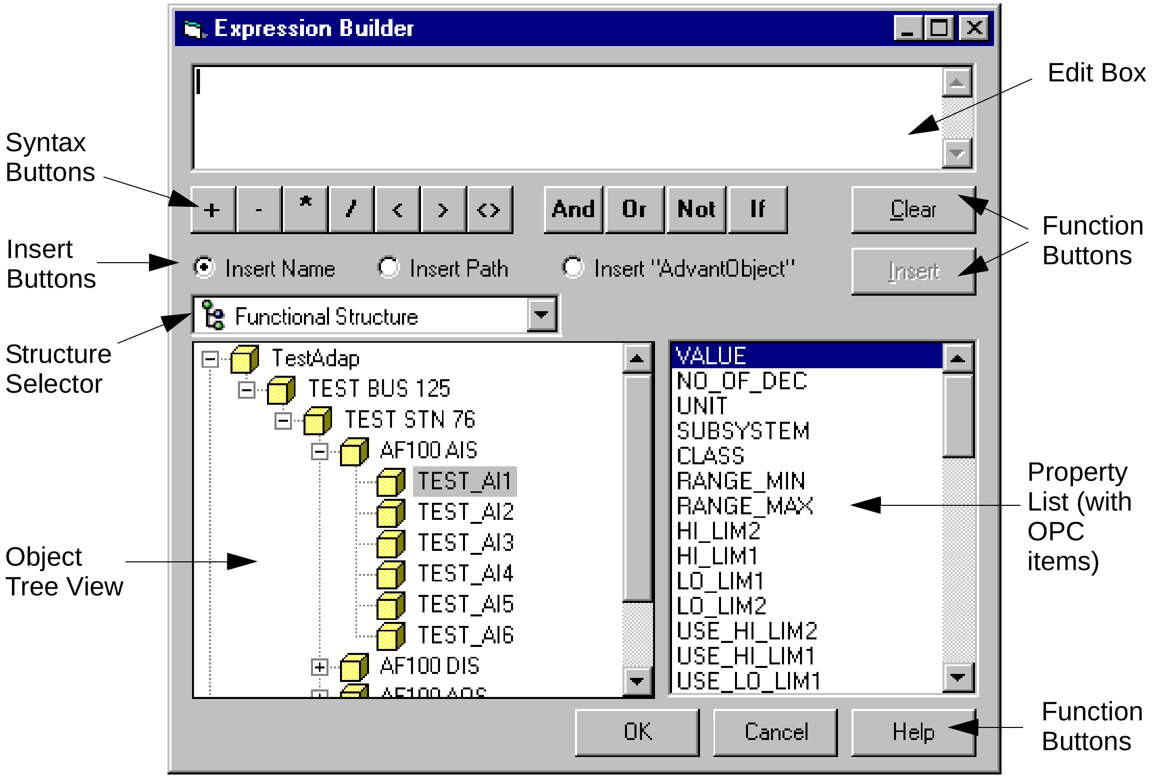


Figure 3-22. Different parts of the Expression Builder window

The sole objective of the Expression Builder is to build and edit expressions, Section 3.16, Building Expressions.

The buttons and fields of the Expression Builder window are:

3.16.1.1 Edit box

The expression is presented in an edit box at the top of the dialog. A few lines are visible and for very long expressions a scroll bar is attached to the right. It is possible to edit the expression manually here.

3.16.1.2 Syntax buttons

Beneath the Edit box is a row of buttons. The purpose of the buttons is to provide help with the syntax of expressions. The symbol or function name on the button is added to the expression in the Edit box when the button is pressed. The text appears at the location of the cursor or replaces a selected portion. If the Edit box is out of focus the text is added to the previous place of the cursor. In most cases, the inserted symbol gets a blank space on both sides. The buttons provide the following symbols and names: +, -, /, *, <, >, <>, And, Or, Not, If.

"If" results in IIf(<Expression>,<TruePart>,<FalsePart>), see also Section B.4, If-then-else Statements.

3.16.1.3 Function buttons

In addition to the syntax buttons there are function buttons performing specific tasks.

Click **Clear** to erase everything from the edit box.

Click **Insert** to add the selected OPC item to the expression.

Click **OK** to close the Expression Builder and pass the expression to the Expression window. If you click **Cancel** instead, the window is closed without passing any expression to the Expression window.

Click **Help** to display the on-line help. The on-line help for the Expression Builder describes the interface of the Expression Builder and the syntax of expressions (see Appendix B, Expressions).

3.16.1.4 Structure Selector

The structure selector is a drop-down list from which you can select the structure to browse.

3.16.1.5 Object Tree View

The area where the selected structure is presented.

3.16.1.6 Property List (with OPC items)

This list shows properties of process control aspects (that is, items in the Advant OPC server) associated with a selected ABB Object. A property is added to the expression by selecting it and pressing the **Insert** button or just by double-clicking the property.

Many process control aspects can have the same property in the list.The Expression Builder identifies the process control aspect to which the property belongs to by adding the following syntax:

[<process control aspect>:<property name>]

For example: [TEST_AI1:NAME]

3.16.1.7 Insert buttons

If more than one ABB Object have the same name, the expression syntax must be able to identify the correct properties (OPC items) in the Advant OPC server. This is solved by adding the path to the ABB Object in the expression.

Use the Insert buttons to automatically add this path.

Select **Insert Name** to add the name of the ABB Object.

Select **Insert Path** to add the path to the ABB Object.

Select **Insert "AdvantObject"** to add the AdvantObject property (see Section 3.17, Relative References).

For more information about the syntax of expressions, see Appendix B, Expressions.

3.17 Relative References

You can design a graphic element or a graphic display to support a structure of ABB Objects, that is, a graphic element with name paths and relative names in the expressions can be used in more than one graphic display if they use an identical structure.

If you want to create relative references, use the relative name **AdvantObject** instead of absolute names and name paths.

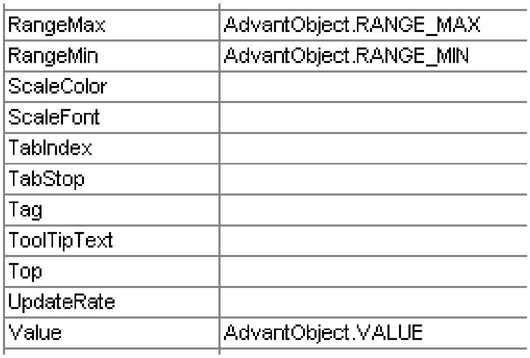
An expression with an absolute name, for example

$'AI.117'.VALUE * 14.2

will look like this[1] with a relative reference

AdvantObject.VALUE * 14.2

which means that the VALUE property of the ABB Object assigned to the AdvantObject property multiplied with 14.2 drives the property to which this expression is applied.



RangeMax	AdvantObject.RANGE_MAX
RangeMin	AdvantObject.RANGE_MIN
ScaleColor	
ScaleFont	
TabIndex	
TabStop	
Tag	
ToolTipText	
Top	
UpdateRate	
Value	AdvantObject.VALUE

Figure 3-23. Example of relative references in the Expressions window

---

[1] You can also use the relative name m_AdvantObject.

SubPath

A SubPath is used to find a “child” object, that is another ABB Object, from the current ABB Object. As child objects exist in most of the typical solutions you probably need to use the SubPaths. SubPaths are used for graphic displays (but not limited to them).

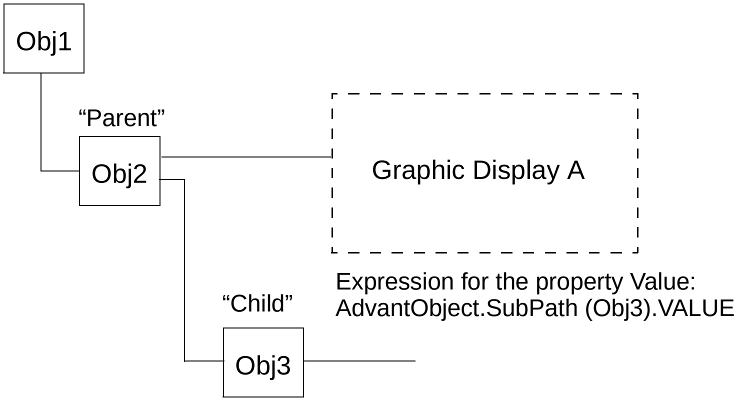

Obj1
“Parent”
Obj2
Graphic Display A
Expression for the property Value:
AdvantObject.SubPath (Obj3).VALUE
“Child”
Obj3

Figure 3-24. SubPath

An expression with a relative reference, for example:

AdvantObject.VALUE

will look like this if you use the subpath:

AdvantObject.SubPath (“Obj3”).VALUE

which means that the VALUE property of another ABB Object (“Obj3”) than the one normally assigned (“Obj2”) drives the property to which this expression is applied.

3.18 Copying Graphic Aspects in the Same Object Structure

If you want to use the same graphic aspect for several ABB Objects it is convenient to copy an aspect which uses relative references. Otherwise you need to change all the expressions for the graphic aspect.

To copy a graphic aspect (with relative references):

1. Make sure that all the expressions of the graphic aspect you want to copy uses the relative name AdvantObject instead of absolute names and name paths.
2. Select the graphic aspect in the structure (in Advant Plant Explorer).
3. Right-click and select **Copy** in the context menu.
4. Select the ABB Object to which you want to copy the graphic aspect.
5. Right-click and select **Paste** in the context menu.

   The copied aspect's layout is identical to the original one's, but this graphic aspect presents different property values.

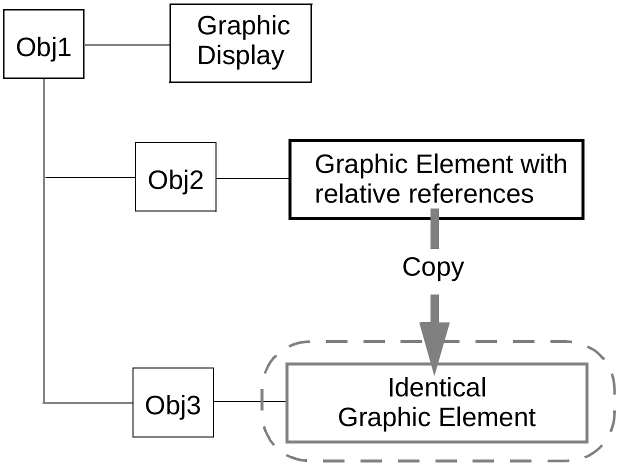

Obj1
Graphic Display
Obj2
Graphic Element with relative references
Copy
Obj3
Identical Graphic Element

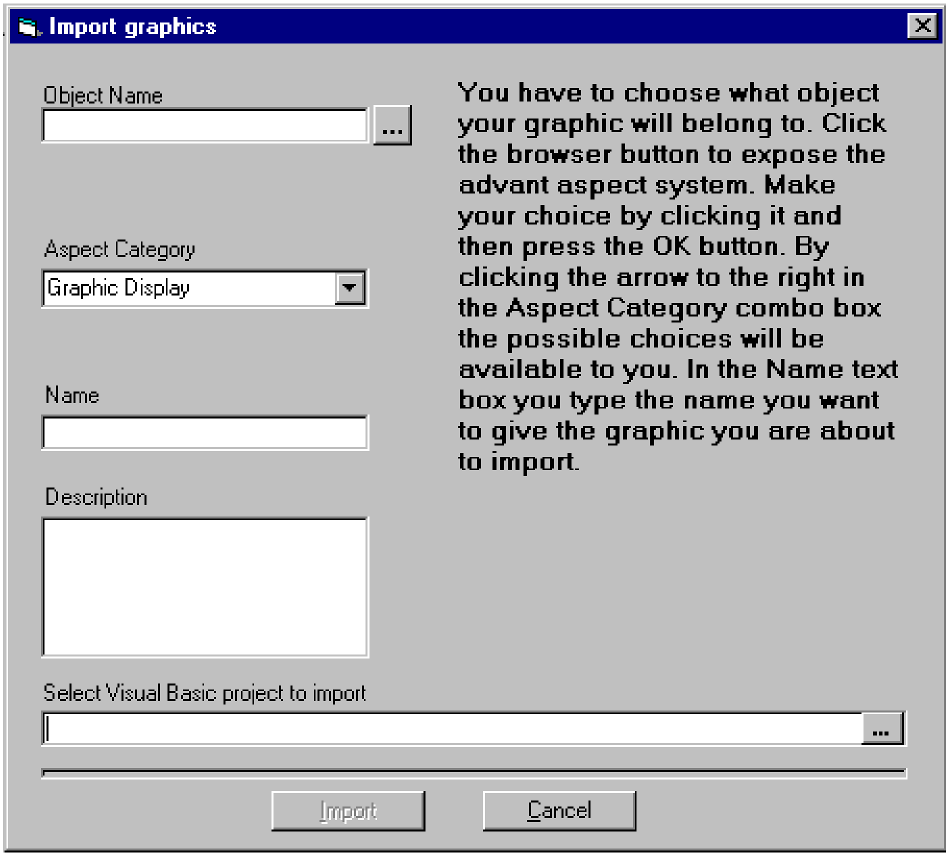
Figure 3-25. Copying graphic aspects with relative references

3.19 Importing Visual Basic Projects

The usual way to create a graphic aspect is to build it (the graphic element) in Advant Graphics Builder. Another way is to build it in Visual Basic and then import the VB project.

To import a Visual Basic project:

1. Open the Import graphics window by choosing **Import** in the File menu.



Aspect Category: Graphic Display

You have to choose what object your graphic will belong to. Click the browser button to expose the advant aspect system. Make your choice by clicking it and then press the OK button. By clicking the arrow to the right in the Aspect Category combo box the possible choices will be available to you. In the Name text box you type the name you want to give the graphic you are about to import.

Figure 3-26. Import graphics window

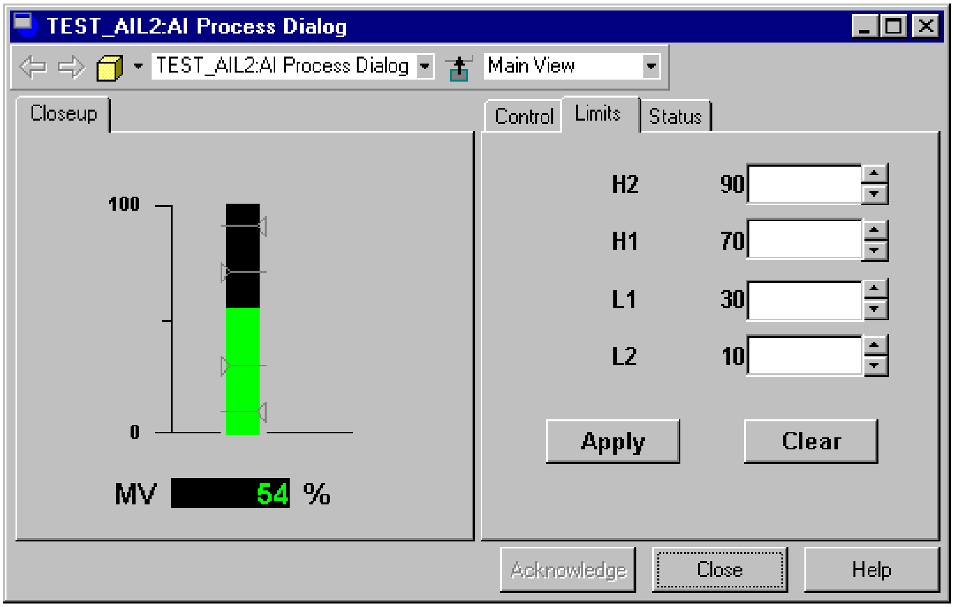
2. Click the Object Name browser button to expose the Advant aspect system (in a separate window).
3. Select the object your graphic will belong to, then press **OK** (which closes the separate window).
4. Click the arrow button to the right in the Aspect Category field and select one of the categories available to you.
5. Enter a name in the Name field (the name you want to give the graphic you are about to import).
6. Click the Visual Basic project browser button to expose the available Visual Basic projects (in a separate window).
7. Select the Visual Basic project that you want to import, then press **OK** (which also closes the separate window).
8. Click **Import**.

The Visual Basic project **must** be a User Control, otherwise it can't be imported.

Chapter 4 How to Configure Process Dialogs

4.1 Introduction

A process dialog is a type of graphic configured from the process dialog window and used for process information and control.



H2 90
H1 70
L1 30
L2 10
MV 54 %

Figure 4-1. Example of a Process Dialog

4.2 Getting Started

The following is a summary of the basic steps for configuring process dialogs from the Configuration View of the process dialog and changing the process dialog user profile.

Configuring the Process Dialog

The basic steps for configuring a process dialog from the Configuration View of the process dialog (see Section 4.3.2, Configuring Process Dialogs) are:

1. Familiarize yourself with the process dialog view by looking through Section 4.2.1, User Interface.
2. Choose structure, object and process dialog in Advant Plant Explorer.
3. Select the Process Dialog aspect and choose View from the context menu. This will open the process dialog window.
4. Choose the Configuration View.
5. Choose one item (tab control).
6. If you want to delete the autoloaded aspects, deselect the Include Auto Load checkbox. The aspects disappear.
7. If you want to add an aspect, click on a cell in the Aspect column and select one of the available aspects in the window that is presented. You can add as many new aspects as you want before you close it.
8. If you want to sort the aspects in an other order, select an aspect and click on the buttons to the left to move it up or down in the list.

   If the autoloaded aspects remain in the list they will automatically be placed in alphabetic order after the manually added aspects.
9. Click **Apply** and select the Main View or Object Display in order to verify the configured process dialog.

Changing the Process Dialog User Profile

The basic steps for changing the process dialog user profile (see Section 4.4.2, Changing the Process Dialog User Profile) are:

1. Familiarize yourself with the process dialog views by looking through Section 4.2.1, User Interface.
2. Choose structure, object and aspect in Advant Plant Explorer.
3. Choose the first item (tab control).

4. Click in the checkbox to the left for each aspect category from which you want to include tabs.
5. To change the order of the aspect categories, select an aspect and click on the buttons to the right to move it up or down in the list.
6. Repeat the procedure (point 3-5) for the second item.
7. If required, click in the checkboxes to select option(s).
8. Click **Apply**.

4.2.1 User Interface

The process dialog window has three views:

- Main Window (see Section 4.3.1.1, Main Window)
- Object Display (see Section 4.3.1.2, Object Display)
- Configuration View (see Section 4.3.1.3, Configuration View).

The first two are used to view the selected aspects for the process dialog and the third is used to change the configuration of the process dialog.

The contents of new process dialog aspects depends on what you have defined for process dialogs in the process dialog user profile in Advant Plant Explorer. Aspects included in the categories you select are automatically presented as tabs in the process dialog - that is, if they exist for the selected ABB Object.

The only time the information in a process dialog is predefined is when you create an ABB Object of a process object type (that is, AIS, AOS, DOS etc.). The process dialog is then created at the same time as you create the new ABB Object.

If you later want to change the configuration of a process dialog, select its Configuration View and make the changes. You can use a combination of automatic and manual configuration, to add specific aspects to the autoloaded ones.

Some features are specific to the process dialog. These are:

- Process dialog user profile
- Configuration View

4.2.2 Application Start-up

The process dialog is accessed by:

- Clicking on an object on the screen (the most common way) or
- Double clicking on the Process Dialog aspect in Advant Plant Explorer.

You access the process dialog user profile from Advant Plant Explorer.

4.3 Working with Process Dialogs

Process dialogs are configured from the Configuration View of the process dialog window. This section gives you a description of the process dialog window, and the Configuration View in particular.

4.3.1 Process Dialog Window

The main objectives of the process dialog window are:

- To present different aspects of an ABB Object.
- To reconfigure the process dialog.

A process dialog contains the following information:

- A number of tabs on the left (upper) and right (lower) side.
- An aspect reference for each tab. These refer to the aspect displayed when the tab is selected.
- A caption for each tab.
- If Operator Log should be displayed or not (Option defined in the process dialog user profile).
- If Apply requires confirmation or not (Option defined in the process dialog user profile).
- If a warning dialog should be displayed if you change tab or closes the dialog if there are un-applied changes (Option defined in the process dialog user profile).

The Process Dialog window has three views which you can switch between with the View Selector:

- Main Window
- Object Display
- Configuration View

4.3.1.1 Main Window

The Main Window gives you a choice of viewing one Information tab and one Control tab. There may be multiple information and control tabs, depending on the selected aspects, but it is only possible to view one tab at a time.

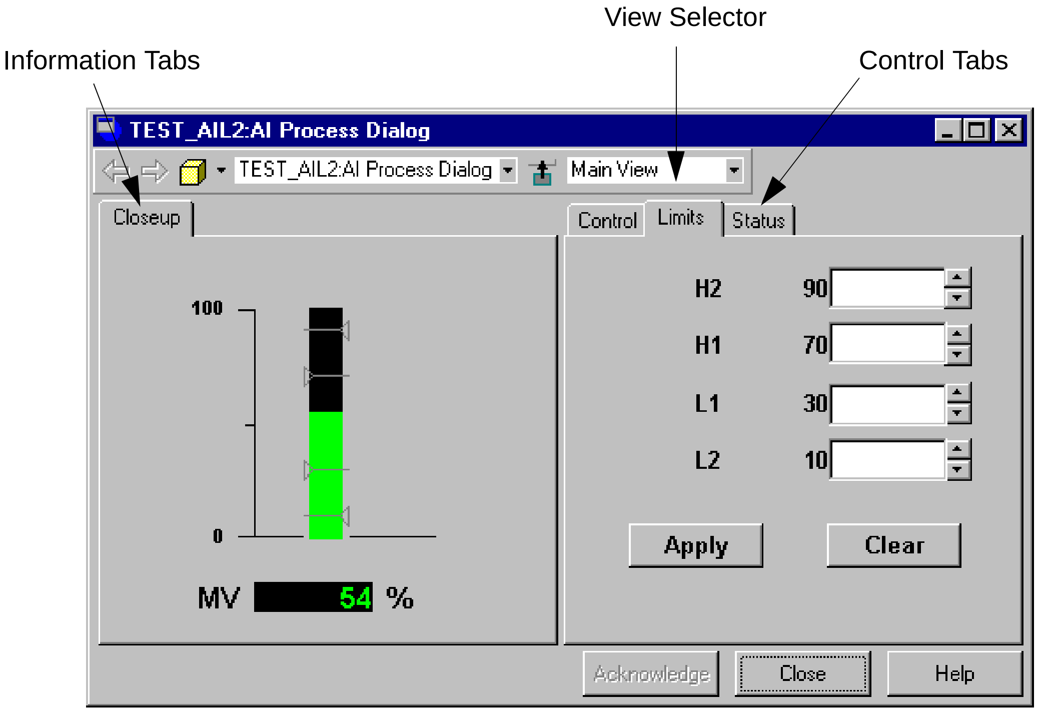


Figure 4-2. Process Dialog - Main Window

The features and buttons specific to the Main Window of the process dialog are:

Tabs

The left tab control normally contains object information, typically close-up and trend aspects. The trend tab(s) shows the properties defined by the trend aspect(s). The trend tab(s) only display the curves and possibly a property list.

The right tab control contains tabs used to control the process by changing process values.

Left and right buttons appear to the right of the tabs if there are more than five tabs above the right tab control.

If one tab control is empty, that is, it contains no aspects, it is not displayed, and the remaining tab control is given all the available space.

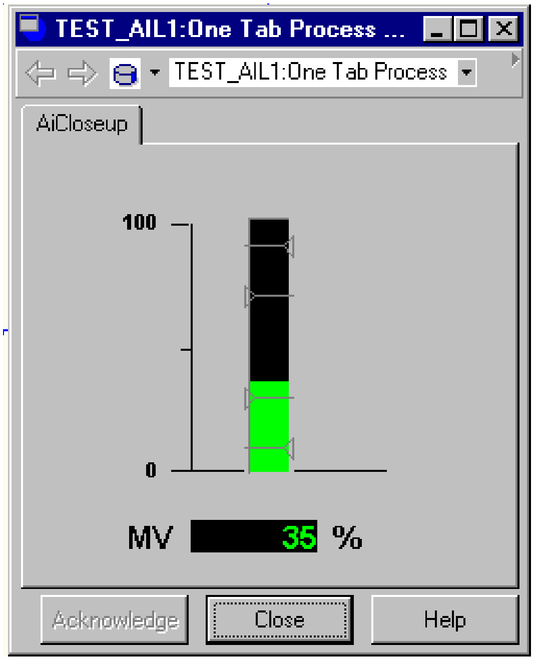


Figure 4-3. Main View with only one tab control

A tab can be read-only or read/write, that is, it contains fields that allow one or more values to be changed by the operator. Read/write tabs can apply changes by direct manipulation (for example with a toggle button), by using the tab buttons Apply and Clear, or using a combination.

Buttons

The **Apply** and **Clear** buttons appear inside the tab control. Use them when you want to apply or delete the values you have entered.

The buttons outside the tab controls applies to the Main Window:

- Click **Acknowledge** to acknowledge all unacknowledged alarms on the object the Main Window is displayed for.
- Click **Close** to close the Main Window.
- The **Help** button shows the help text for the process dialog.

The Acknowledge button is only visible if you have the required privilege.
The Close button is only visible if the process dialog is displayed in an overlapping window.

4.3.1.2 Object Display

In the Object Display, all Information and Control tabs are visible at the same time. The tabs are laid out in frames in two rows, one for Information tabs and one for Control tabs.

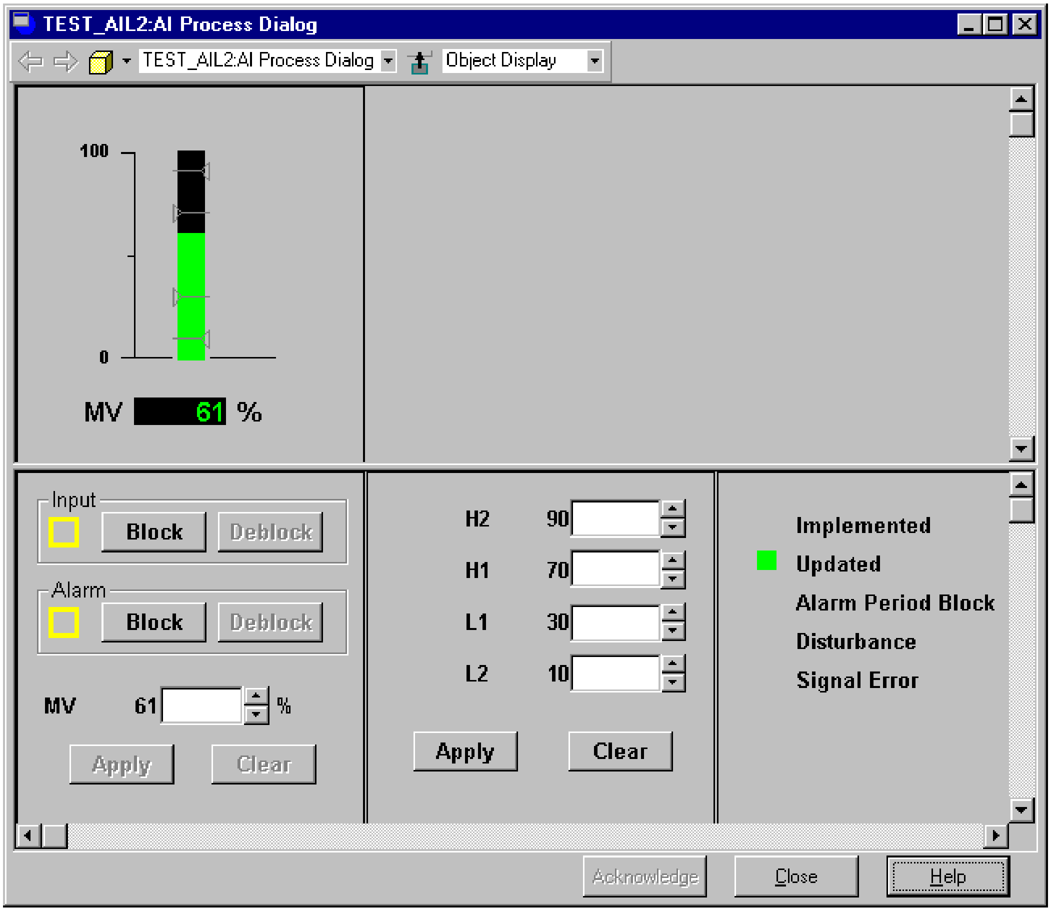


Figure 4-4. Process Dialog - Object Display

If the tabs do not fit inside the window, vertical and horizontal scrollbars are added.

The default size of the Object Display is the same as a base display. This default size applies when the object displayed is viewed as an overlapping display, otherwise the object display will adapt to the size of its frame.

The features and buttons specific to the Object Display of the process dialog are:

Tabs

The top row of the Object Display contains object information, typically from the left a close-up and to the right trend aspects. The trend tab(s) shows the properties defined by the trend aspect(s). The trend tab(s) only display the curves and possibly a property list. The bottom row of the Object Display contains tabs used to control the process by changing process values.

Scrollbars appears at the bottom of the Object Display and to the right of the rows.

A tab can be read-only or read/write, that is, it contains fields that allow one or more values to be changed by the operator. Read/write tabs can apply changes by direct manipulation (for example with a toggle button), by using the tab buttons Apply and Clear, or using a combination.

Buttons

The **Apply** and **Clear** buttons appear inside the tab control. Use them when you want to apply or delete the values you have changed for a tab.

The buttons outside the tab controls applies to the Object Display:

- Click **Acknowledge** to acknowledge all unacknowledged alarms on the object the Object Display is displayed for.
- Click **Close** to close the Object Display.
- The **Help** button shows the help text for the process dialog aspect.

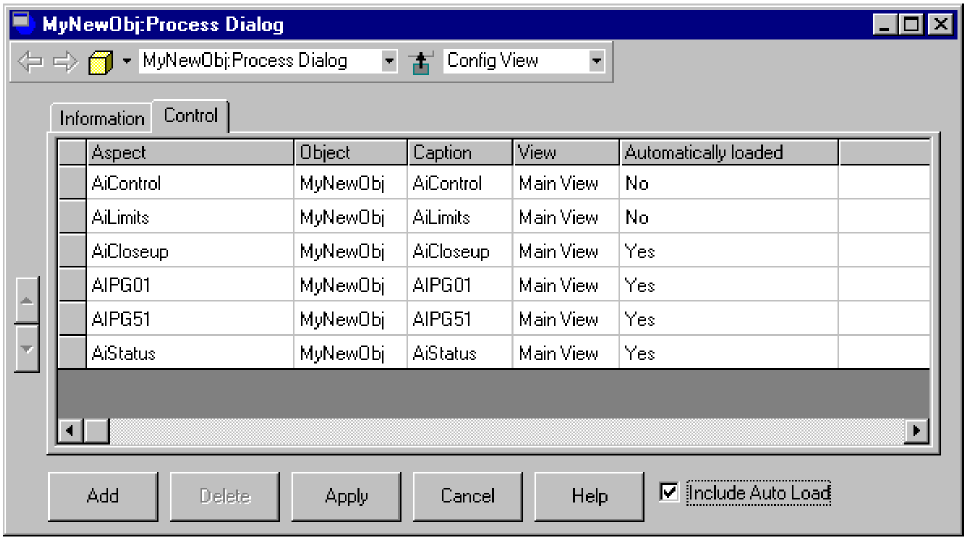
The Acknowledge button is only visible if you have the required privilege.
The Close button is only visible if the process dialog is displayed in an overlapping window.

4.3.1.3 Configuration View

This view is only used when you want to configure the two other views.

The Configuration View is only available if you have the required privilege.

The Configuration View shows the list of all aspects that will be available in the other two views. In this view, it is possible to modify the contents of the list. In addition to the listed aspect data, the configuration view also displays what aspect view will be used, and if the aspect is automatically loaded or not.



Aspect	Object	Caption	View	Automatically loaded
AiControl	MyNewObj	AiControl	Main View	No
AiLimits	MyNewObj	AiLimits	Main View	No
AiCloseup	MyNewObj	AiCloseup	Main View	Yes
AIPG01	MyNewObj	AIPG01	Main View	Yes
AIPG51	MyNewObj	AIPG51	Main View	Yes
AiStatus	MyNewObj	AiStatus	Main View	Yes

☑ Include Auto Load

Figure 4-5. Process Dialog - Configuration View

You can use:

- Manual Configuration
- Automatic Configuration (Auto Load)

It is also possible to use a combination of automatic and manual configuration, to add specific aspect to the automatically loaded ones (as in the figure above).

The features and buttons specific to the Configuration View of the process dialog are:

Tabs

Two tabs are available, one for information aspects and one for control aspects. Information about the loaded aspects are listed below each tab.

Tab Lists

The list includes the aspect name, the name of the ABB Object to which it belongs, the aspect tab caption, the selected process dialog view and information whether the aspect has been automatically loaded.

To add or change an aspect, click on a cell in the Aspect column and select one of the available aspects in the window that is presented.

Checkboxes

Only the aspects from the aspect categories you selected in the process dialog user profile are automatically loaded. The autoloaded aspects are visible only as long as the box Include Auto Load is checked.

Up/Down Buttons

If you want to change the order of the aspects, select an aspect and click on the buttons to the right to move it up or down in the list.

You can not change the order of autoloaded aspects. They are alphabetically listed in a group.

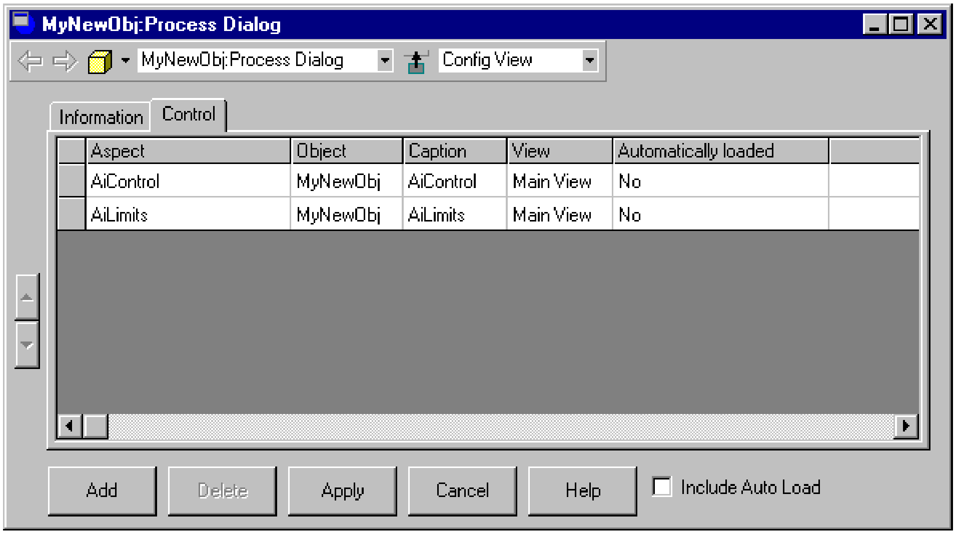
Buttons

The buttons below the tab list are:

- **Add**, to add an aspect to the list.
- **Delete**, to delete an aspect from the list.
- **Apply**, to apply changes in the list.
- **Cancel**, to cancel changes in the list.
- **Help**, to show the help text for the Configuration View.

Manual Configuration

When manually configured, the listed aspect data is defined in the Configuration View of the process dialog aspect. Normally, most tabs will be Graphic Element aspects.



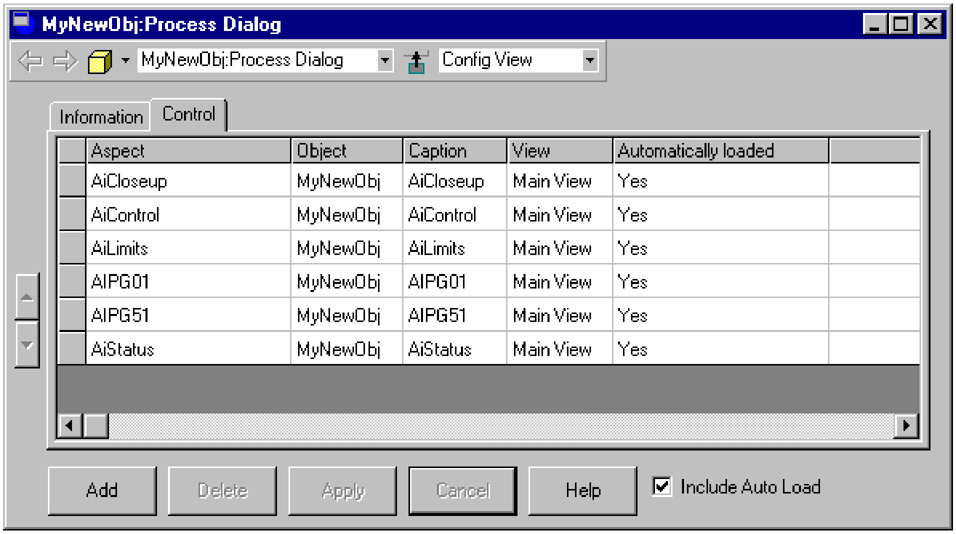
Aspect	Object	Caption	View	Automatically loaded
AiControl	MyNewObj	AiControl	Main View	No
AiLimits	MyNewObj	AiLimits	Main View	No

☐ Include Auto Load

Figure 4-6. Configuration View - Manual Configuration

Automatic Configuration (Auto Load)

For automatic configuration, the process dialog user profile defines the aspect categories that are automatically included in the Main Window and Object Display views (see Section 4.4.2, Changing the Process Dialog User Profile). All aspects of the selected categories will be included.



Aspect	Object	Caption	View	Automatically loaded
AiCloseup	MyNewObj	AiCloseup	Main View	Yes
AiControl	MyNewObj	AiControl	Main View	Yes
AiLimits	MyNewObj	AiLimits	Main View	Yes
AIPG01	MyNewObj	AIPG01	Main View	Yes
AIPG51	MyNewObj	AIPG51	Main View	Yes
AiStatus	MyNewObj	AiStatus	Main View	Yes

☑ Include Auto Load

Figure 4-7. Configuration View - Automatic Configuration

It is possible to use a combination of automatic and manual configuration, to add specific aspect to the automatically loaded ones.

The autoloaded aspects are visible in the three views as long as the box **Include Auto Load** is checked.

4.3.2 Configuring Process Dialogs

Process Dialogs can be manually or automatically configured.

The Configuration View (see Section 4.3.1.3, Configuration View) shows the list of all aspects that will be available in the other two views. In this view, it is possible to modify contents of the list. In addition to the aspect data listed above, the Configuration View also displays what aspect view will be used, and if the aspect is automatically inserted or not.

When manually configured, the aspect data listed above is defined in the Configuration View of the process dialog aspect. Normally, most tabs will be Graphic Element aspects.

For automatic configuration (see Section 4.4.2, Changing the Process Dialog User Profile), the process dialog user profile defines the aspect categories that should automatically be included in the process dialog/object display configuration. All aspect of the selected categories will be included.

It is also possible to use a combination of automatic and manual configuration, to add specific aspect to the automatically loaded ones.

The Configuration View is only available if you have the required privilege.

To configure the process dialog from the Configuration View:

1. Choose the Object Type structure in Advant Plant Explorer.
2. Choose the Process Dialog of an ABB Object.
3. Select the Process Dialog aspect and choose View from the context menu. This will open the process dialog window.
4. Choose the Configuration View with the View Selector.
5. Choose one item (tab control).
6. If you want to delete the autoloaded aspects, deselect the Include Auto Load checkbox. The aspects disappear.
7. If you want to add an aspect, click on a cell in the Aspect column and select one of the available aspects in the window that is presented. You can add as many new aspects as you want before you close it.
8. If you want to sort the aspects in an other order, select an aspect and click on the buttons to the left to move it up or down in the list.

   If the autoloaded aspects remain in the list they will automatically be placed in alphabetic order after the manually added aspects.
9. Click Apply and select the Main View or Object Display.

4.4 Working with the Process Dialog User Profile

Process dialogs can initially be configured from the process dialog user profile in Advant Plant Explorer. This section gives you a description of the process dialog user profile and how you configure it.

4.4.1 Process Dialog User Profile

Enter the process dialog user profile in Advant Plant Explorer by choosing Aspect System Structure, the ABB Object "Process Dialog" and the aspect "Process Dialog Profile Definition..

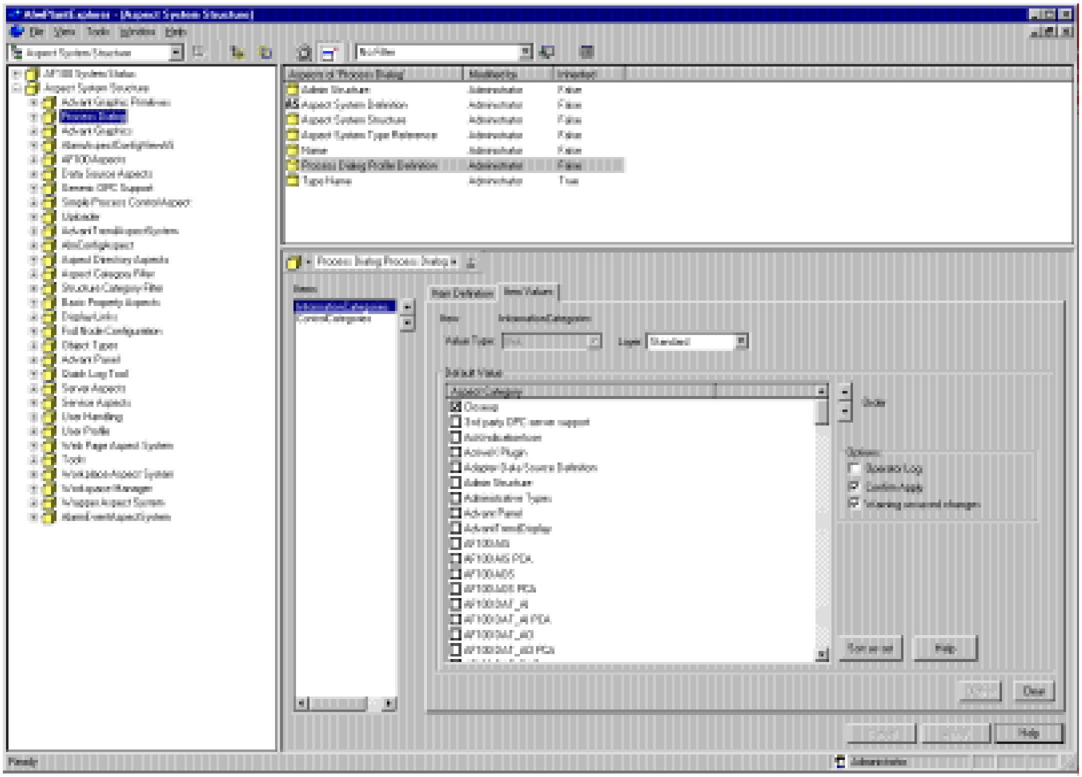


Figure 4-8. Advant Plant Explorer with the Process Dialog User Profile selected

4.4.1.1 Process Dialog Profile Definition

The aspect "Process Dialog Profile Definition" contains:

Aspect Selector

The aspect selector shows the ABB Object "Process Dialog" which you have selected. You can use it to change the aspect of the ABB Object.

Items List

Includes the two items InformationCategories and ControlCategories. Each item has a list of available aspect categories in the Aspect Category list.

Aspect Category List

The Aspect Category list contains all the available aspect categories for the selected item. To select aspect categories from which you want to include aspect tabs, click in the checkboxes to the left.

Up/Down Buttons

Use these buttons to change the order of the aspect categories. Select an aspect category in the Aspect Category list and click on the up or down button to move it in the list.

Options

The Options part contains three alternatives: Operator Log, Confirm Apply, and Warning unsaved changes. To select an option, click in the checkboxes to the left.

Choose **Confirm Apply** if Apply requires confirmation or not. The "Confirm Apply" function means that the process dialog "ask" the "tab aspect" if it has been changed. The reason is that the process dialog can not detect whether the "tab aspect" has been changed or not.

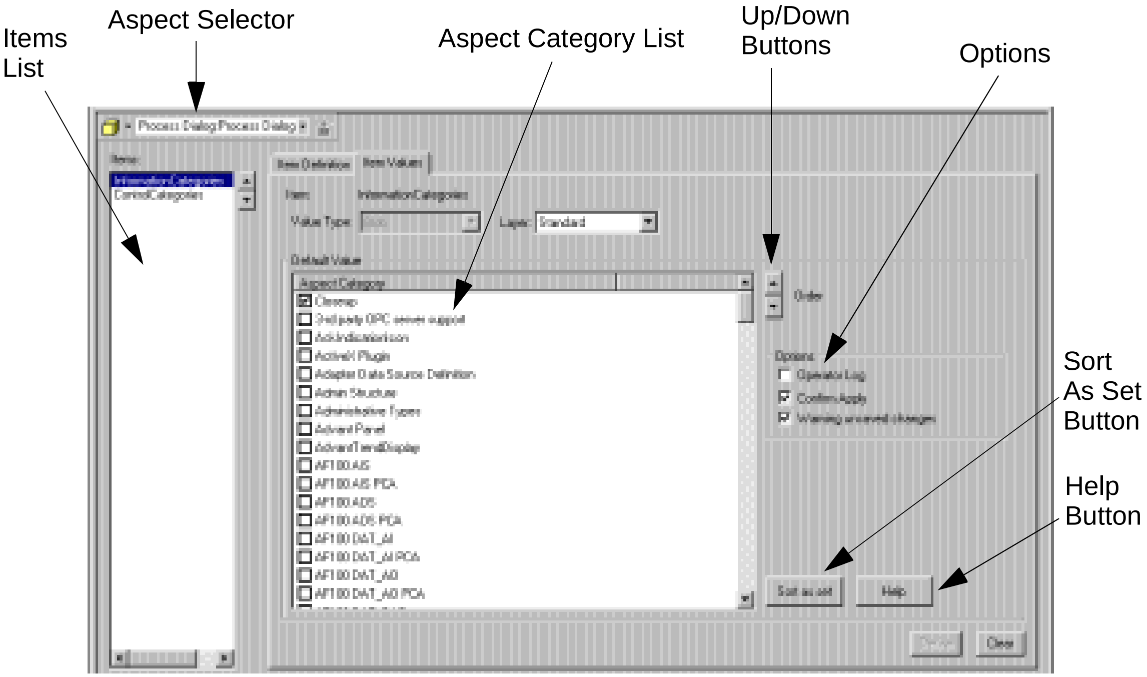
Choose **Warning unsaved changes** if you want a warning before you close a process dialog with unsaved changes. A warning dialog is then presented if you change a tab's control values or close the process dialog without having applied the changes. The "Warning unsaved changes" function means that the process dialog "ask" the "tab aspect" if it has been changed. The reason is that the process dialog can not detect whether the "tab aspect" has been changed or not.

Sort as set Button

If you want to sort the selected aspect categories as a set, click **Sort as set**.

Help Button

If you want the help text for the Process Dialog Profile Definition, click **Help**.



Item: InformationCategories

Layer: Standard

Options
☐ Operator Log
☑ Confirm Apply
☑ Warning unsaved changes

Figure 4-9. Process Dialog User Profile

4.4.2 Changing the Process Dialog User Profile

To change the process dialog user profile:

1. Start Advant Plant Explorer.
2. Choose **Aspect System Structure**
   (Note! This changes the default value for all users!).
3. Choose the ABB Object **Process Dialog**
   (Note! This changes the default value for all users!).
4. Choose the aspect **Process Dialog Profile Definition**
   (Note! This changes the default value for all users!).
5. In the items list, select **InformationCategories**.
6. In the Aspect Category list, click in the checkbox to the left for each aspect category from which you want to include information tabs.

   An aspect category in the list does not necessarily represent just one tab in the process dialog. For example, a category for a primitive element or subelement provided by ABB can automatically create several tabs.
7. If you want to change the order of the information aspect categories select a category and click on the buttons to the right to move it up or down in the list. The aspects within each category is sorted alphabetically (default) when the process dialog is presented.
8. In the items list, select **ControlCategories**.
9. In the Aspect Category list, click in the checkbox to the left for each aspect category from which you want to include control tabs.
10. If you want to change the order of the control aspect categories, select an aspect and click on the buttons to the right to move it up or down in the list.
11. If you want to select one or more of the options (Operator log, Confirm Apply, Warning unsaved changes), click in the checkboxes for the options.
12. Click **Apply**.

Every process dialog you view which have not been re-configured will now show aspects of the same aspect categories - that is, if they exist for the selected ABB Object.

Chapter 5 How to Build Graphic Displays

5.1 Introduction

A graphic display is an aspect of an ABB Object, which consist of static and dynamic graphic objects that provides both orientation and live process feedback.

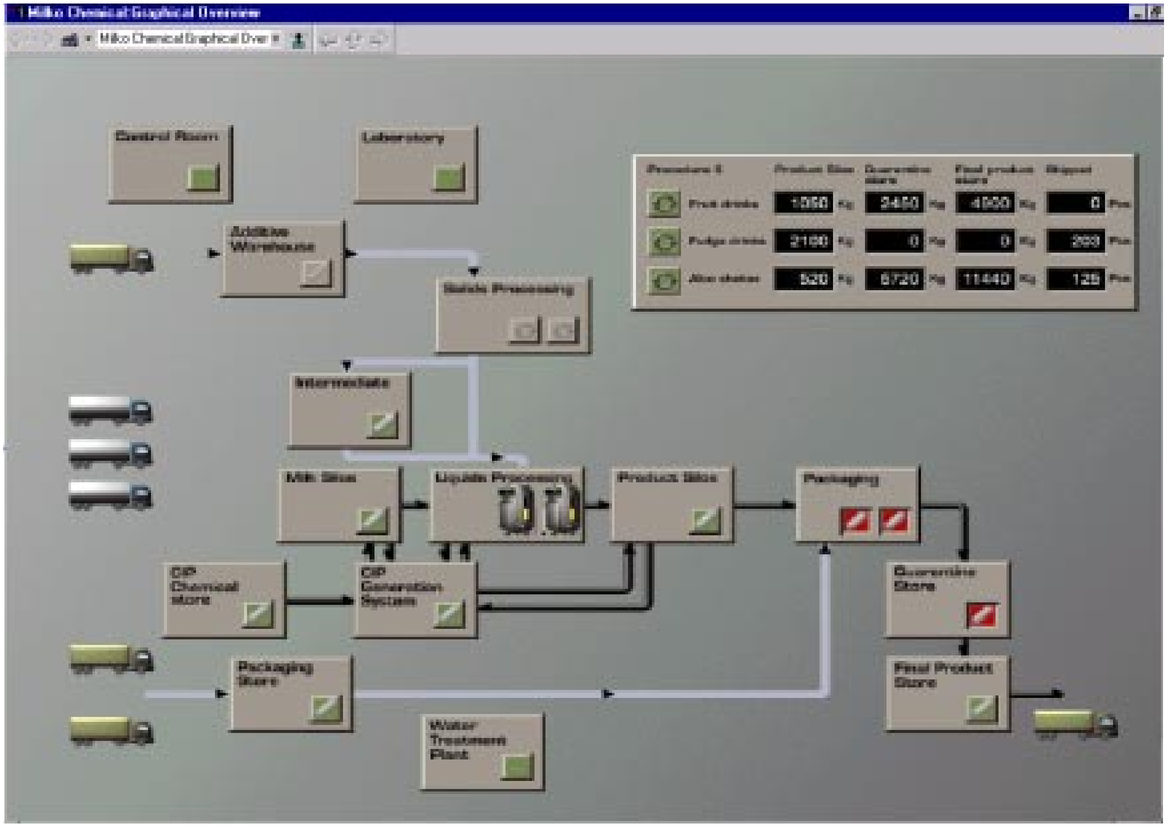


Figure 5-1. Example of a Graphic Display

5.2 Getting Started

The following is a summary of the basic steps for using Advant Graphics Builder when you are working with graphic displays.

1. Familiarize yourself with Advant Graphics Builder by looking through Section 5.2.1, User Interface.
2. Choose structure, object and aspect in Advant Plant Explorer.
3. Right-click the aspect you want and choose **Edit** from the context menu. This will start Advant Graphics Builder, Section 5.2.2, Application Start-up.

   If this is a new graphic display a wizard will help you to get the right settings to get started, Section 3.3.1, Advant Graphics Builder Window[1].
4. Advant Graphics Builder is now ready to be used. Select the tools you need from the toolbox, Section 3.10, Working with the Toolbox[1].
5. Build/edit the graphic display, Section 5.4, Building Graphic Displays and Section 5.5, Editing Graphic Displays.
6. Test the graphic displays, Section 5.6, Testing Graphic Displays
7. Save the graphic displays before you exit the application, Section 5.7, Saving Graphic Displays.
8. Deploy the graphic displays, Section 5.8, Deploying Graphic Displays.

5.2.1 User Interface

Advant Graphics Builder has most of Visual Basic's features for building graphic displays, plus some specific to this system. These are:

- The Advant Graphics wizard that guides you through the steps of building graphic displays.
- The Design and Test function that enables you to build graphic displays in design mode and then check their behaviour in test mode.
- The Add Tool function that allows you to add tools to your toolbox.

---

[1] The sections Section 3.3.1, Advant Graphics Builder Window and Section 3.10, Working with the Toolbox ALSO applies when you are working with graphic elements (see Chapter 3, How to Build Graphic Elements).

- The Graphic Element Browser in which you can search for appropriate graphic elements.
- The Deploy function which stores the graphic displays and makes them accessible for viewing. Before a graphic display is deployed it can only be seen in Advant Graphics Builder.
- The Help function that offers you access to On-line Help and On-line Documentation for Advant Graphics Builder.

These features are accessed through the pull-down menu in the menu bar or through special buttons.

5.2.2 Application Start-up

To start Advant Graphics Builder you can either create a new aspect or edit an existing one.

Create a New Aspect

1. Start Advant Plant Explorer.
2. Select a structure and an ABB Object.
3. Right-click in the aspect area and choose **New Aspect** from the context menu.
4. In the New Aspect window: Select aspect category in the structure list (Aspect System structure/Advant Graphics/Graphic Display/Graphic Display), give the new aspect a name and click **Create**.
5. Select the new aspect in the Aspect information list, right-click and choose **Edit** from the context menu. Advant Graphics Builder and its wizard starts. (You have to follow some of the wizard's steps before you can begin building the graphic display.)

Edit an Existing Aspect

1. Select the aspect (graphic display).
2. Right-click and choose **Edit** from the context menu. Advant Graphics Builder starts with the selected graphic display.

5.3 Working with Advant Graphics Builder

Advant Graphics Builder provides several features for building and editing graphics (graphic elements/displays). The Advant Graphics Builder window is described in Section 3.3.1, Advant Graphics Builder Window.

When you are working with graphic displays, the main objectives of Advant Graphics Builder are:

- To build graphic displays. Section 5.4, Building Graphic Displays.
- To edit graphic displays. Section 5.5, Editing Graphic Displays.
- To test graphic displays. Section 5.6, Testing Graphic Displays.
- To save graphic displays. Section 5.7, Saving Graphic Displays.
- To deploy graphic displays. Section 5.8, Deploying Graphic Displays.

Several sections in Chapter 3, How to Build Graphic Elements, ALSO applies when you are working with graphic displays:

- Section 3.3.1.1, Advant Graphics Wizard
- Section 3.3.1.2, Graphic Element Browser
- Section 3.3.1.3, Add Tool
- Section 3.3.2, Setting Advant Options
- Section 3.10, Working with the Toolbox
- Section 3.14.1, Graphic Element Browser Window
- Section 3.17, Relative References
- Section 3.18, Copying Graphic Aspects in the Same Object Structure
- Section 3.19, Importing Visual Basic Projects

One section in this chapter ALSO applies when you are working with graphic elements:

- Section 5.10, Scaling Bitmap Pictures

5.4 Building Graphic Displays

There are two ways in which you can start building graphic displays:

- Starting from Advant Plant Explorer
- Starting from Advant Graphics Builder

5.4.1 Starting from Advant Plant Explorer

Below follows a description of how to build a graphic display when you start from Advant Plant Explorer.

1. Start Advant Plant Explorer.
2. Select a structure.
3. Select an ABB Object.
4. Add an aspect by right-clicking in the aspect area and choose **New Aspect** from the context menu. The New Aspect window is presented.
5. Select the aspect category Graphic Display in the structure list (Aspect System structure/Advant Graphics/Graphic Display/Graphic Display).
6. Give the new aspect a name and click **Create**. The selected aspect is added to the Aspect information list.
7. Right-click it and choose **Edit** from the context menu. This starts the Advant Graphics wizard in Advant Graphics Builder. Page Three in the wizard is presented. Select the tools you want to add (can also be done later).
8. Click **Finish**. It is now possible to start building a graphic display.

   Notice that a display workspace covers a much larger area than an element workspace and that they use different background colors (green and grey).
9. Build your graphic display with the help of the tools (ActiveX controls) in the toolbox and the graphics you can find through the Graphic Element Browser. The Symbol Factory is also useful when building displays.

5.4.2 Starting from Advant Graphics Builder

Below follows a description of how to build a graphic display when you start from Advant Graphics Builder.

1. Either click the Wizard button in the toolbar or choose **New Aspect** in the File menu. This starts the Advant Graphics wizard in Advant Graphics Builder. Page One in the wizard is presented.
2. Click **Next**. Page Two in the wizard is presented.
3. Click on the browser button for the Object Name field and select an ABB Object.
4. Click on the Object Name field’s Browser button. The Object Browser window is presented.
5. Select the ABB Object name and click **OK**.
6. Then you select the aspect category Graphic Display from the combo box.
7. Give your graphic display a name.
8. Enter a description text (optional).
9. Click **Next**. Page Three in the wizard is presented.
10. Click **Finish**. It is now possible to start building a graphic display (in the green area).
11. Build your graphic display with the help of the tools (ActiveX controls) in the toolbox and the graphics you can find through the Graphic Element Browser.

5.5 Editing Graphic Displays

There are two ways in which you can start editing graphic displays:

- Starting from Advant Plant Explorer
- Starting from Advant Graphics Builder

5.5.1 Starting from Advant Plant Explorer

Below follows a description of how to edit a graphic display when you start from Advant Plant Explorer.

1. Select the ABB Object and the graphic aspect (that is, the graphic display) which you want to edit.
2. Right-click on the aspect in the aspect area. A context menu appears.
3. Choose **Edit** from the menu. Advant Graphics Builder now starts with the selected graphic display.
4. Edit the graphic display.
5. Save your changes.

5.5.2 Starting from Advant Graphics Builder

Below follows a description of how to edit a graphic display when you start from Advant Graphics Builder.

1. Either click the Open button in the toolbar or choose **Open Aspect** in the File menu. The Open Aspect window is presented.
2. Select the ABB Object and the graphic aspect (that is, the graphic display) which you want to edit.
3. Either click **Open** or double-click on the selected aspect.

Another way, which is even faster, is to use the drag-and-drop function. Simply drag the selected aspect to the white area of the Visual Basic's workspace.

4. Edit the graphic display.
5. Save your changes.

5.6 Testing Graphic Displays

Graphic displays are built in Design mode. However, the data subscriptions are not set up, expressions are not executed, controls are not animated, etc. To check this you use the Test mode.

The Test mode provides functionality to:

- Check behavior of a graphic display.
- Check that all object references are resolved.

Hence, when you decide to switch from Design mode to Test mode your graphic display will set up a subscription, and values, animation, etc. are continuously updated.

To test a graphic display:

1. Switch to test mode by clicking the Test Mode button on the toolbar, or choose **Test Mode** under the Run menu.
2. Check the behavior of the graphic display.
3. Click the Design Mode button on the toolbar, or choose **Design Mode** under the Run menu to return to design mode.

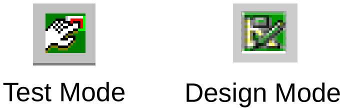


Figure 5-2. Test Mode and Design Mode Buttons

Using AF100 Test OPC Server (the off-line data source)

Advant Graphics contains a way to test your graphical elements, process dialogs and process graphics by using the off-line data source, the AF100 Test OPC Server.

See the *Advant Workplace Products Administrators Guide*, chapter 3 for information on how to connect the OPC server and how to use the Uploader aspect.

5.7 Saving Graphic Displays

To save a graphic display you either click the Save button in the toolbar or choose **Save** from the File menu. You do not have to save the graphic in any particular location. The system decides where to store all graphics.

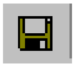


Figure 5-3. Save Button

5.8 Deploying Graphic Displays

To deploy a graphic display you choose **Deploy** from the File menu. The Deploy function is used to make it possible to view a graphic display in the Advant Workplace Window. If a graphic display is not deployed you can only see it in Advant Graphics Builder.

5.9 Deleting Graphic Displays

To delete a graphic display, right-click it in Advant Plant Explorer and choose **Delete** from the context menu.

5.10 Scaling Bitmap Pictures

Graphic displays built with the standard template in Advant Graphics Builder, where you use a bitmap picture (or gif or jpeg) as background, may not be correctly scaled to fit within the base display area of the Advant Workplace Window. This may cause misinterpretation of information and faulty control actions. This is avoided if you make the bitmap picture scalable.

To make a bitmap picture (as background in a graphic display) scalable:

1. Create a new graphic display in Advant Graphics Builder.
2. Click the icon for the **Image** tool (ActiveX control) in the toolbox (under the General button).
3. Place the control in Visual Basic's workspace.

4. Resize the control so that it covers the whole workspace.
5. Open the Properties window.
6. Select the **Picture** property.
7. Click on the button at the far right in the property's edit area. The Load Picture window is presented.
8. Search for the desired bitmap picture, select it and click **Select**. The bitmap picture appears within the control in the workspace.
9. Select the **Stretch** property.
10. Set the value to True.

Chapter 6 How to View Graphics in Run-time

6.1 Introduction

The main viewing window enables you to view object details in graphics (graphic elements and graphic displays) and respond to changes in them. This is useful if you are building a graphic and want to view the result, or if you are observing and supervising a process in a factory. All graphics are shown in run-time. This ensures that all values shown where there are data subscriptions are actual values.

6.2 Getting Started

The following is a summary of the basic steps for viewing object details:

1. Select a graphic to view.
2. View the object detail and if it is necessary, respond to it.

6.3 Viewing Graphics

Viewing graphics is not a separate application. It is only used by Advant Workplace Window to display graphics and starts when you select one of the views of a graphic aspect.

There are three ways that the graphics can be presented. These are listed below with the most common at the top.

- Start Advant Workplace Window. Select Advant Plant Explorer via the Tools Menu. Double-click on the aspect.
- Select the aspect in Advant Plant Explorer. Right-click and select **Main View** in the context menu.
- Select a graphic aspect in any tool capable of presenting objects/aspects. Drag and drop the aspect on the **Drop Target** button in the Advant Workplace Window.

If you want to edit the graphic you are viewing, right-click and select Edit in the context menu. This takes you to Advant Graphics Builder where you can edit the graphic (see Section 3.5, Editing Graphic Elements and Section 5.5, Editing Graphic Displays). This action is only possible if you have configuration privileges.

6.3.1 Main Viewing Window

The main viewing window is a graphic presentation of the selected element.

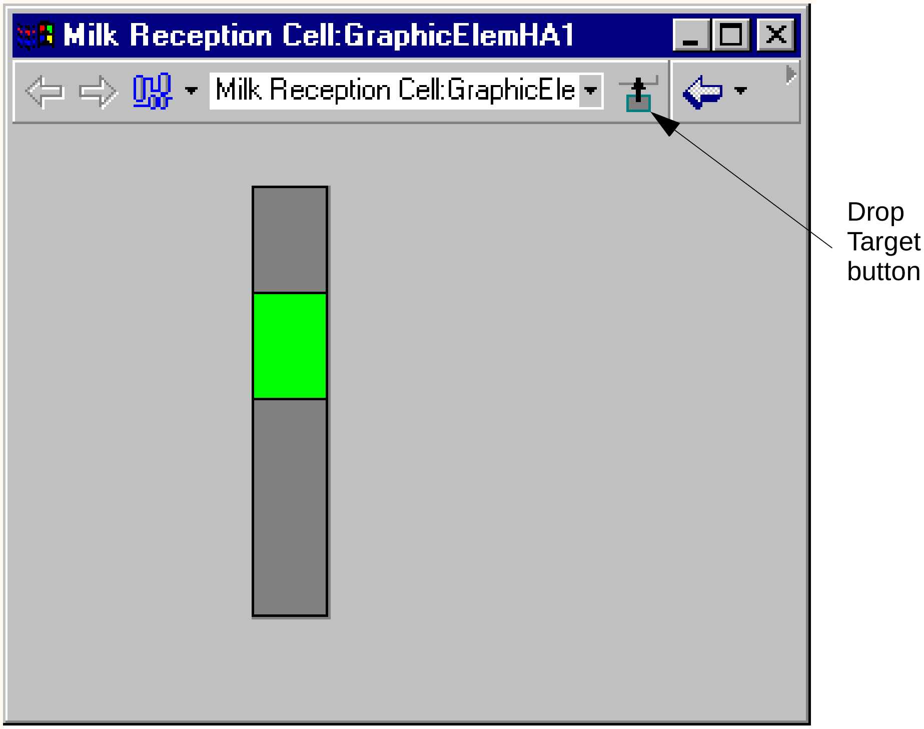

Milk Reception Cell:GraphicElemHA1
Milk Reception Cell:GraphicEle
Drop
Target
button

Figure 6-1. Main Viewing Window

The main objectives of the window are:

- To view graphics. Section 6.3, Viewing Graphics
- To respond to changes in graphics. Section 6.5, Respond to Changes in Graphics

6.4 Object Marking

A graphic element in a graphic display which is connected to an ABB Object is highlighted when you place the cursor over it.

6.5 Respond to Changes in Graphics

The graphics you view can handle dynamic data. If you have a data subscription to an object the data is shown in real-time. A right-click on an graphic element in the graphic display presents a context menu of actions appropriate to that object, that is, if the object is an ABB Object. You can also respond to changes by entering new values or using buttons in the process dialog.

Chapter 7 Maintenance

7.1 Preventive Maintenance

Advant Graphics is a part of the Advant Workplace Products software package. Perform any required maintenance according to the instructions in the *Advant Workplace Products Administrator's Guide*.

7.2 Frequently Asked Questions (FAQ)

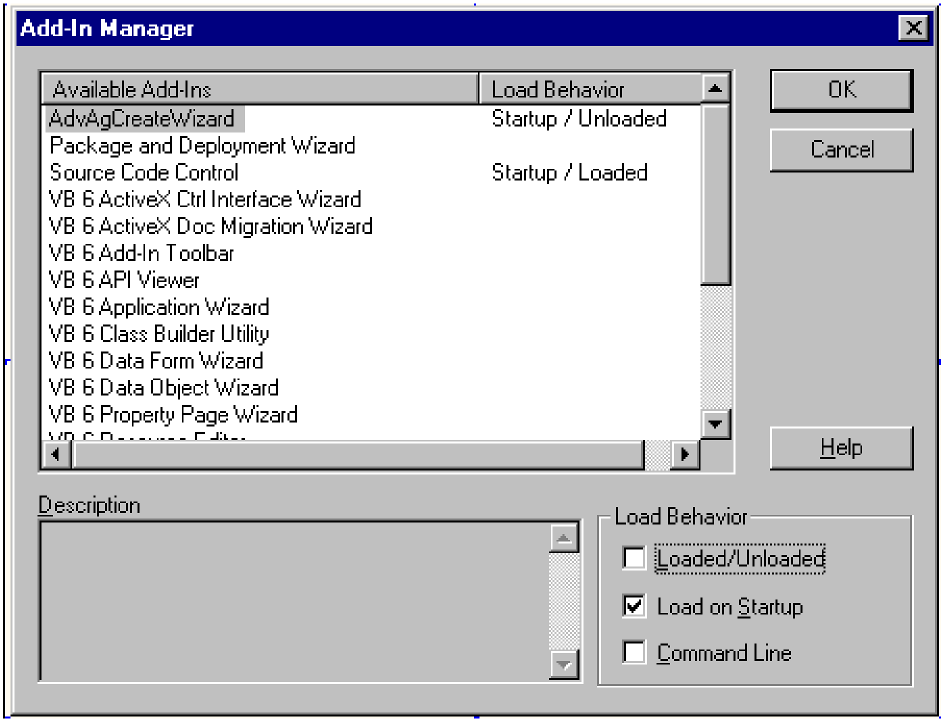
Method '~' of object '~' failed

I get the message Method '~' of object '~' failed. when deploying a graphic element. What is wrong?

It's probably a compile error. Try Test Mode or disconnect the wizard in Advant Graphics Builder by using **Add-ins > Add-In Manager**.

To disconnect the wizard and get an error description:

1. Select **Add-ins > Add-In Manager**. The Add-In Manager window is presented.
2. Select **AdAgCreateWizard** in the list.
3. Deselect the **Loaded/Unloaded** checkbox (the box should be empty).
4. Click **OK** to close the Add-In Manager window.
5. Select **File > Make *aspectID*.ocx**.
6. A dialog box text describing the error is presented.



Available Add-Ins	Load Behavior
AdvAgCreateWizard	Startup / Unloaded
Package and Deployment Wizard	
Source Code Control	Startup / Loaded
VB 6 ActiveX Ctrl Interface Wizard	
VB 6 ActiveX Doc Migration Wizard	
VB 6 Add-In Toolbar	
VB 6 API Viewer	
VB 6 Application Wizard	
VB 6 Class Builder Utility	
VB 6 Data Form Wizard	
VB 6 Data Object Wizard	
VB 6 Property Page Wizard	

Load Behavior

☐ Loaded/Unloaded
☑ Load on Startup
☐ Command Line

Figure 7-1. Add-In Manager

Chapter 8 Tutorial

Below follows a description of how to build, test, save, deploy and use a new graphic element.

Two elements are created. One (Bar Element) using Expression Builder to add functionality, and the other (Valve Element) using Visual Basic code to add functionality. An object called Graphic Library will be created in the tutorial under the functional structure.

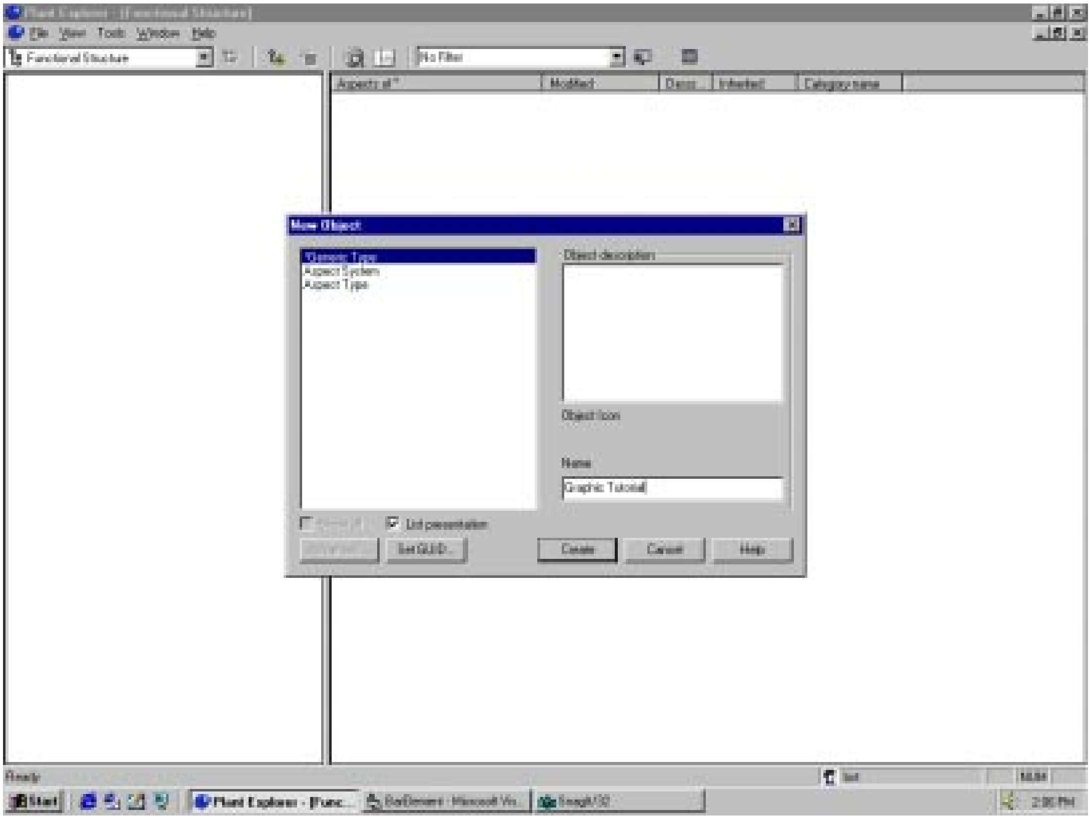
The tutorial consist of the following sections:

- Creating a Graphic Element from Advant Plant Explorer
- Building a Graphic Element
- Changing Properties for the Graphic Element
- Assigning Expressions to the Graphic Element
- Building Expressions for the Graphic Element
- Testing the Graphic Element
- Saving a Graphic Element
- Deploying a Graphic Element
- Using the Graphic Element in a Graphic Display
- Creating a Graphic Element and Using Visual Basic Code to Animate It

8.1 Creating a Graphic Element from Advant Plant Explorer

To create a new graphic element, start from Advant Plant Explorer.

1. Select the **Functional Structure** with the structure selector in Advant Plant Explorer.
2. Add the ABB Object **Graphic Tutorial** in the structure to hold the graphic elements. The object type is ***Generic Type**. Select **Create**.



Name: Graphic Tutorial

☑ List presentation

Figure 8-1. Adding an ABB Object in the functional structure

3. Add an aspect by right-clicking in the Aspect list and choose **New Aspect** from the context menu. The New Aspect window is presented.
4. Select the aspect category **Graphic Element** in the structure list.

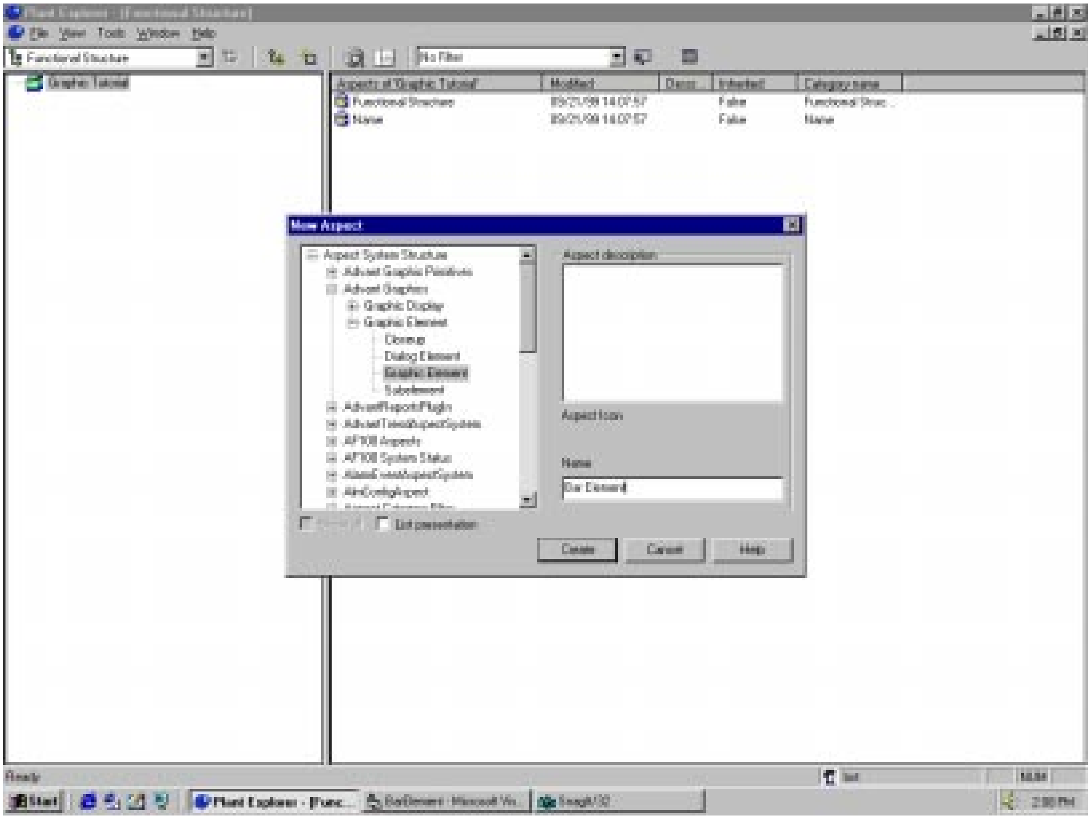
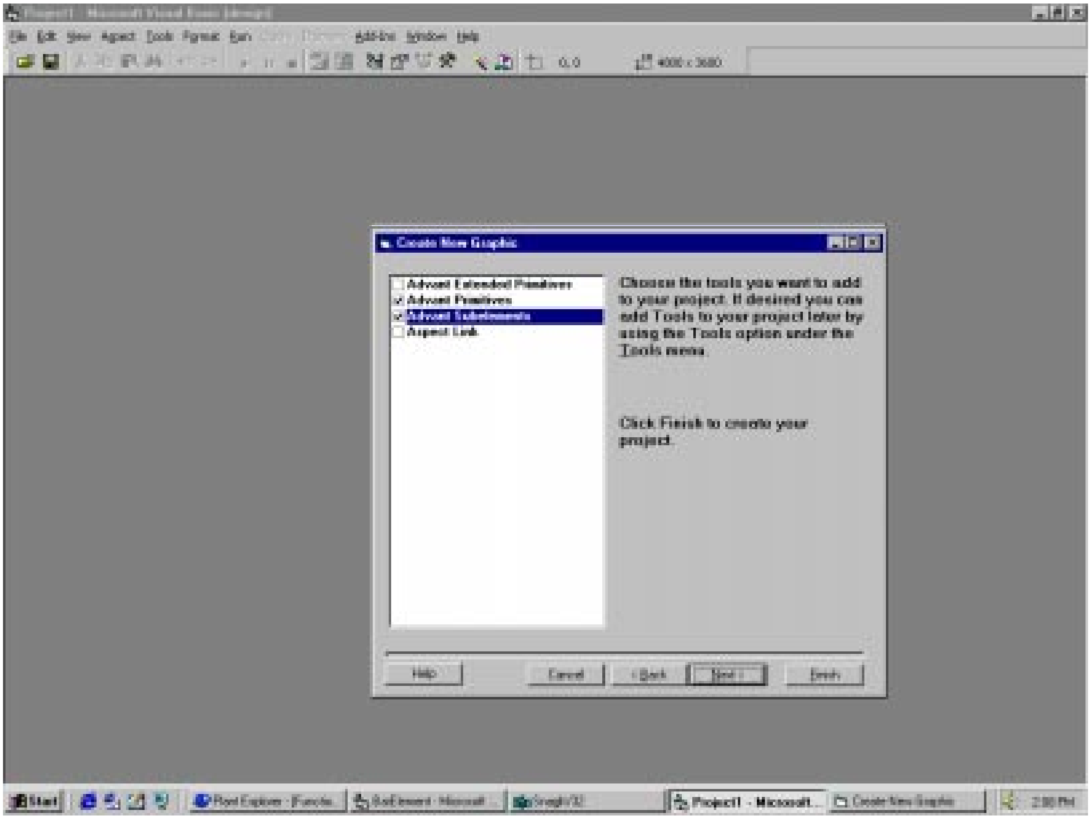


Figure 8-2. Selecting aspect category and adding an aspect

5. Give the new aspect the name **Bar Element** and click **Create**. The selected aspect is added to the Aspect list.
6. Right-click it and choose **Edit** from the context menu. This starts the wizard in Advant Graphics Builder. Page Three in the wizard is presented.



Create New Graphic

- ☐ Advant Extended Primitives
- ☑ Advant Primitives
- ☑ Advant Subelements
- ☐ Aspect Link

Choose the tools you want to add to your project. If desired you can add Tools to your project later by using the Tools option under the Tools menu.

Click Finish to create your project.

Figure 8-3. Starting the Advant Graphics Builder wizard

7. If you want to use a listed tool which isn't already selected, click in the appropriate box in the list. To add additional tools later, see Section 3.10.5, Adding Tools to the Toolbox.

   For the Bar Element, we will not add members (This means that we don't have to continue to the next page in the wizard). This will be done for the valve example (see Section 8.10, Creating a Graphic Element and Using Visual Basic Code to Animate It).

8. Select **Finish** to begin editing the element.

8.2 Building a Graphic Element

To build the new graphic element:

9. Click on the icon for the desired graphic element in the toolbox. In our example, it is the **AdvantBar** in the Advant Subelements section.
10. Place the cursor at any point in the Visual Basic's workspace.
11. Click and hold the left mouse button.
12. Drag the cursor diagonally to the opposite corner of the bounding rectangle.
13. Release the mouse button. The selected graphic element now appears inside a bounding rectangle in the workspace.

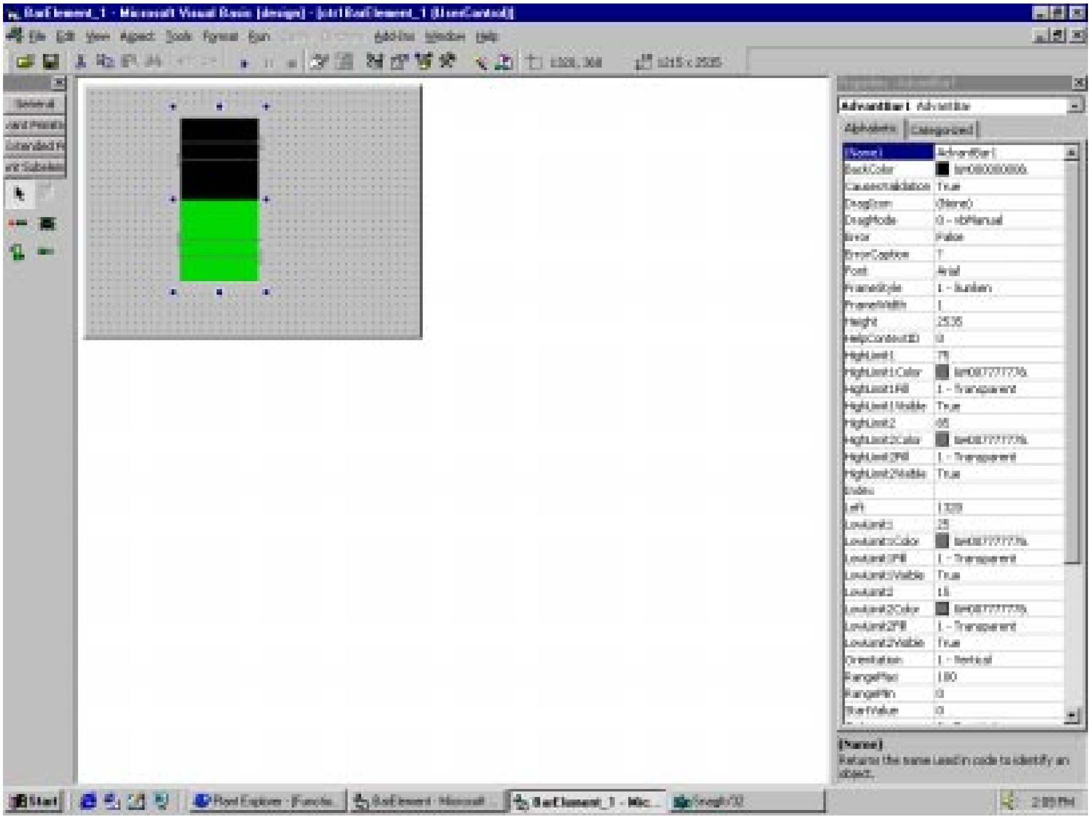


Figure 8-4. A graphic element in the Visual Basic's workspace

14. Continue adding to the graphic element until it is complete. Use the toolbox and the Element Browser window to add other building blocks to your graphic element.

    To use the toolbox, see Section 3.10.4, Providing Elements from the Toolbox. Refer to Section 3.14, Searching for Graphic Elements and Adding Them to a Graphic for using the Element Browser to add System Objects.

8.3 Changing Properties for the Graphic Element

To change the properties BackColor and FrameWidth of the new graphic element:

15. Select the graphic element **AdvantBar** in the workspace.
16. Select the **Properties Window** in the **View** menu. The Properties window is presented.
17. Click in the right column for FrameWidth and change the value to 2.
18. Click in the right column for BackColor and select a yellow color from the palette.

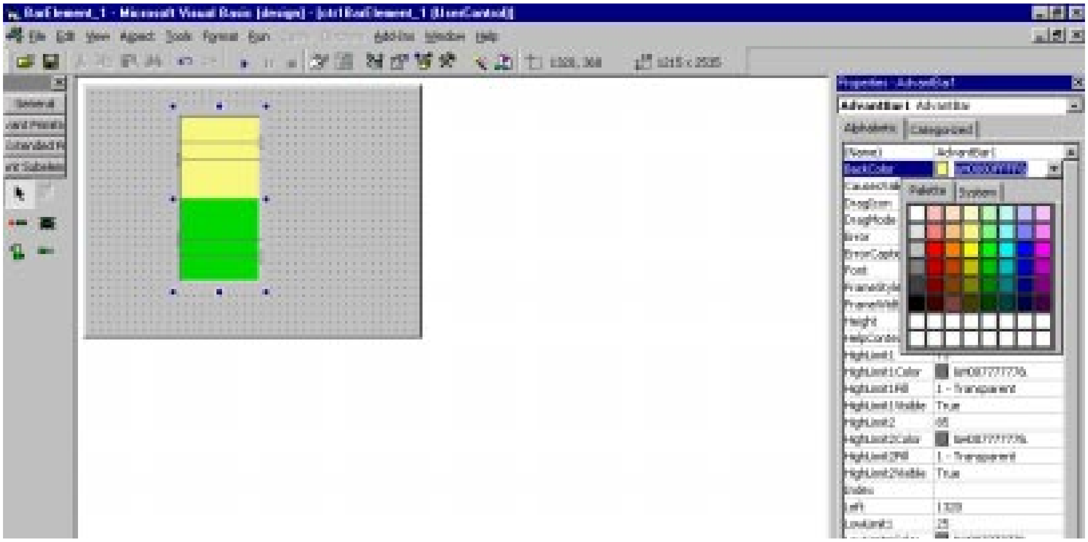


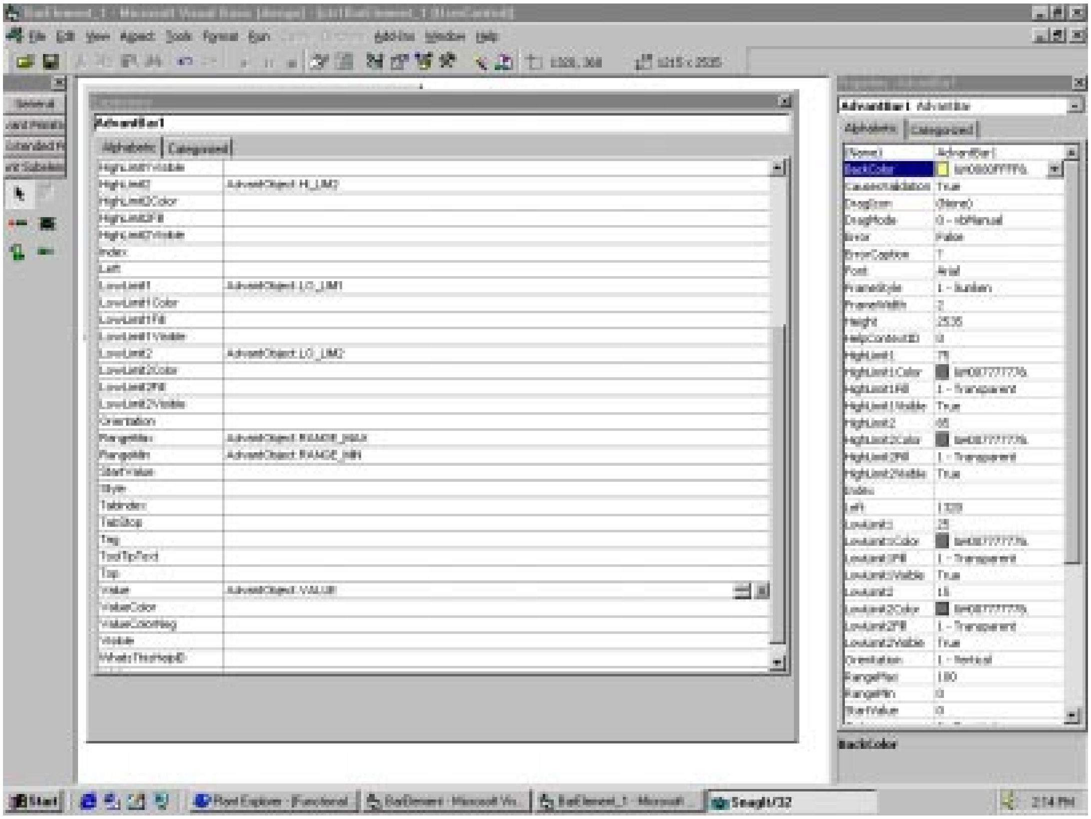
Figure 8-5. Changing properties for a graphic element

8.4 Assigning Expressions to the Graphic Element

All properties can be assigned an expression. Expressions can either be written directly in the Expressions window or generated by the Expression Builder.

To assign an expression to a property directly in the Expressions window:

19. Select the graphic element **AdvantBar** in the workspace.
20. Select **Expression Window** in the **View** menu. The Expressions window is presented. (If necessary: Resize/move the Expression window.) In the figure below, some expressions with relative references (they include AdvantObject) have been entered.



AdvantBar1

HighLimit2	AdvantObject.HI_LIM2
LowLimit1	AdvantObject.LO_LIM1
LowLimit2	AdvantObject.LO_LIM2
RangeMax	AdvantObject.RANGE_MAX
RangeMin	AdvantObject.RANGE_MIN
Value	AdvantObject.VALUE

Figure 8-6. Entering expressions in the Expressions window

AdvantObject refers to the object whose name is displayed by the ObjectName property when the graphic element is used in future displays.

8.5 Building Expressions for the Graphic Element

You can also build an expression in the Expression Builder window and then assign it to the property. The example in this section is not part of the overall tutorial and it should not affect the later sections. To build and assign expressions in the Expression Builder window:

21. Click in the right column of a property in the Expressions window.
22. Click on the "equal" button (=) at the far right of edit area. The Expression Builder window is presented.
23. Build the expression by entering it in the edit box and clicking **OK**.

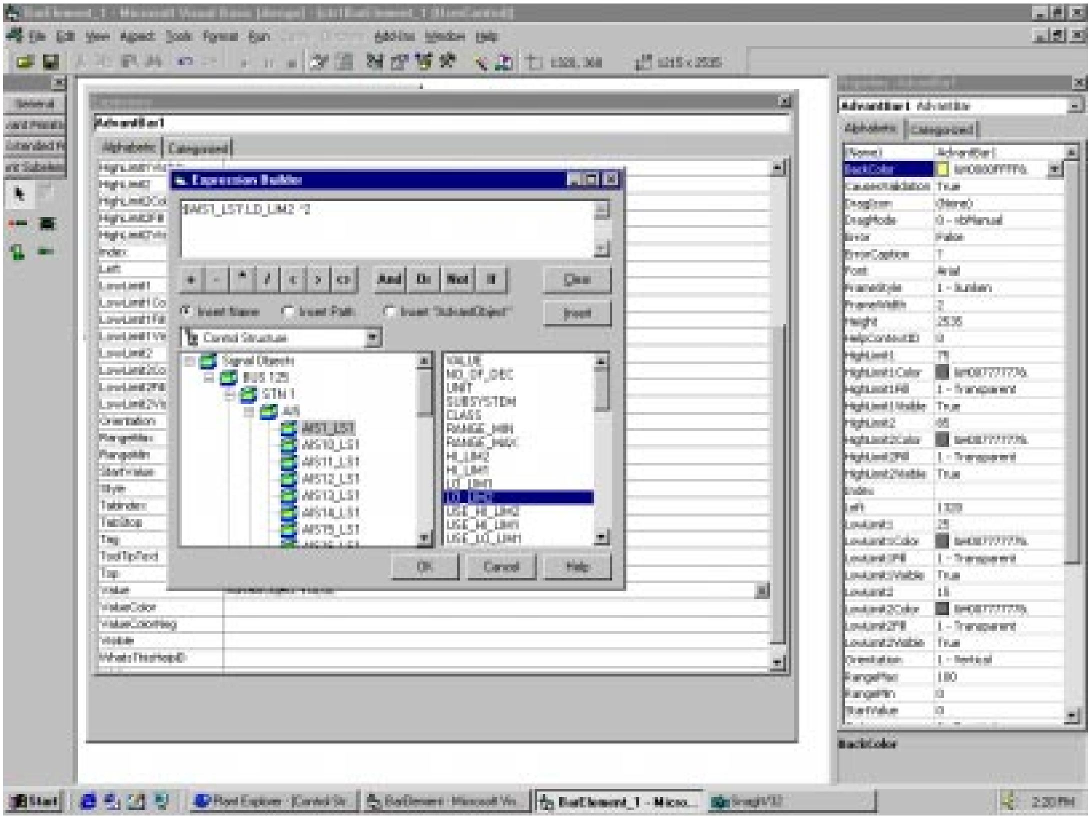


Figure 8-7. Building expressions in the Expression Builder window

The expression in the Figure 8-7, `$’AIS1_LS1’.LO_LIM2 * 2` means that the property LO_LIM2 for the ABB Object AIS1_LS1 is to be multiplied by 2.

You can write everything directly in the edit box, insert ABB Object names or name paths from the Object Tree View and Property List, insert the relative name AdvantObject, as well as add symbols and names with the function buttons (+, -, /, *, <, >, <>, And, Or, Not, If).

To build the expression in the figure:

a. Select the **Control Structure**.

b. Open the ABB Object ‘**Signal Objects, BUS 125, STN 1, AIS, AIS1_LS1**’ in the Object Tree View.

c. Select the value **LO_LIM2**, click Name in the Insert selector, then click **Insert**.

24. Click **OK** to close the Expressions window.

8.6 Testing the Graphic Element

In the above example, we have used the AdvantObject as a reference in the Expressions window. These references need an object which we will provide when we insert the graphic element into a graphic display by assigning the ObjectName property the name of an object.

To test the new graphic element:

25. Select the graphic element **Bar Element**.
26. Switch to test mode by clicking the Test Mode button on the toolbar, or choose **Test Mode** in the **Run** menu.
27. Set the ObjectName property by selecting an AIS system object (for example TEST_AIL1).

If you haven't memorized the name of the AIS system object, do as follows:

Search for the appropriate object in Advant Plant Explorer, select and open it's Name aspect, copy the name and paste it into the ObjectName property field.

28. Check the behaviour for the graphic element.
29. Click on the Design Mode button on the toolbar, or choose **Design Mode** in the **Run** menu to return to design mode.

8.7 Saving a Graphic Element

To save the new graphic element:

30. Click the Save button in the toolbar or choose **Save** from the **File** Menu.

8.8 Deploying a Graphic Element

To deploy the new graphic element:

31. Choose **Deploy** from the **File** menu

The Deploy function makes it possible to view the graphic element in the Advant Workplace Window. If a graphic element is not deployed, you can only see it in Advant Graphics Builder.

8.9 Using the Graphic Element in a Graphic Display

Next we will create a simple graphic display with our Bar Element.

32. If you haven't already done it: Save and deploy the Bar Element.

33. Create a new aspect which is of type **Graphic Display**, call it **Test Display** and click **Create**.

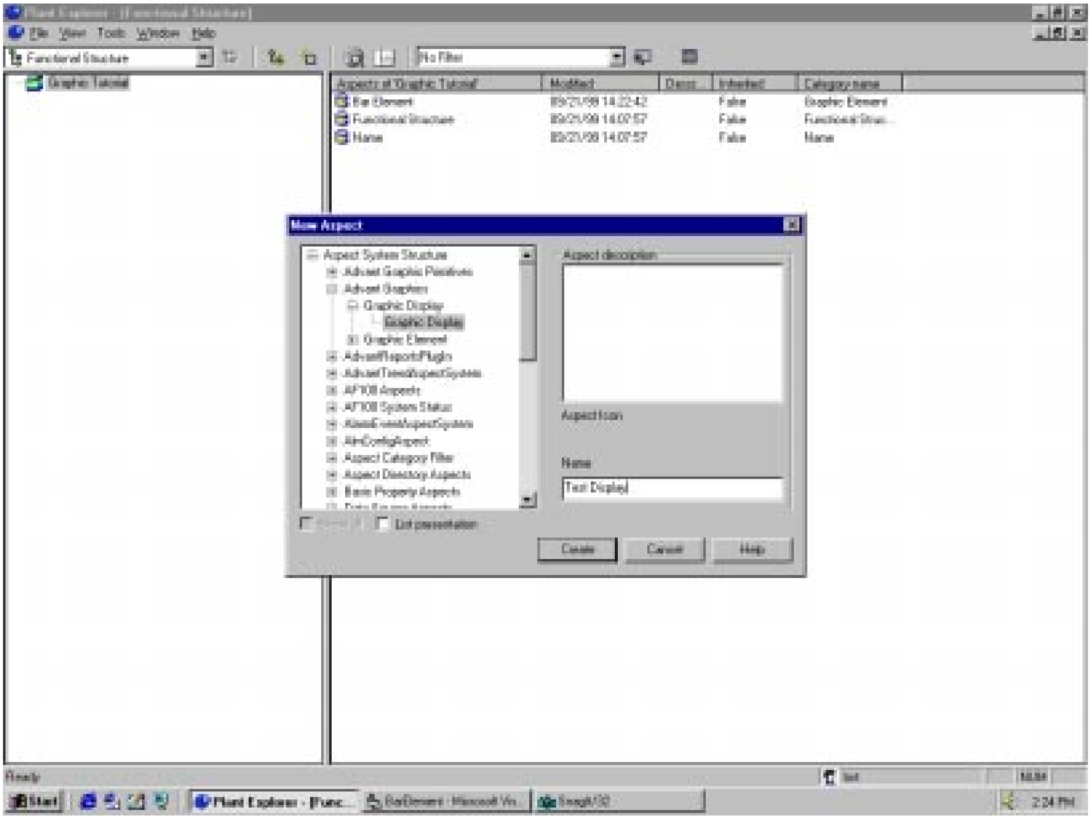


Figure 8-8. Adding another ABB Object in the functional structure

34. Right-click the new aspect and choose **Edit** from the context menu to start the Advant Graphics Builder wizard. Page Three in the wizard is presented.
35. Select the Advant Primitives and Advant Subelements, and click **Finish**.
36. To add our new Bar Element, select **Element Browser** from the **Tools** Menu.

37. Navigate to the Graphics Tutorial in the Functional Structure and select it. You should now see our new Bar Element aspect in the right window.
38. Select the **Bar Element**, and click **Add Element**.

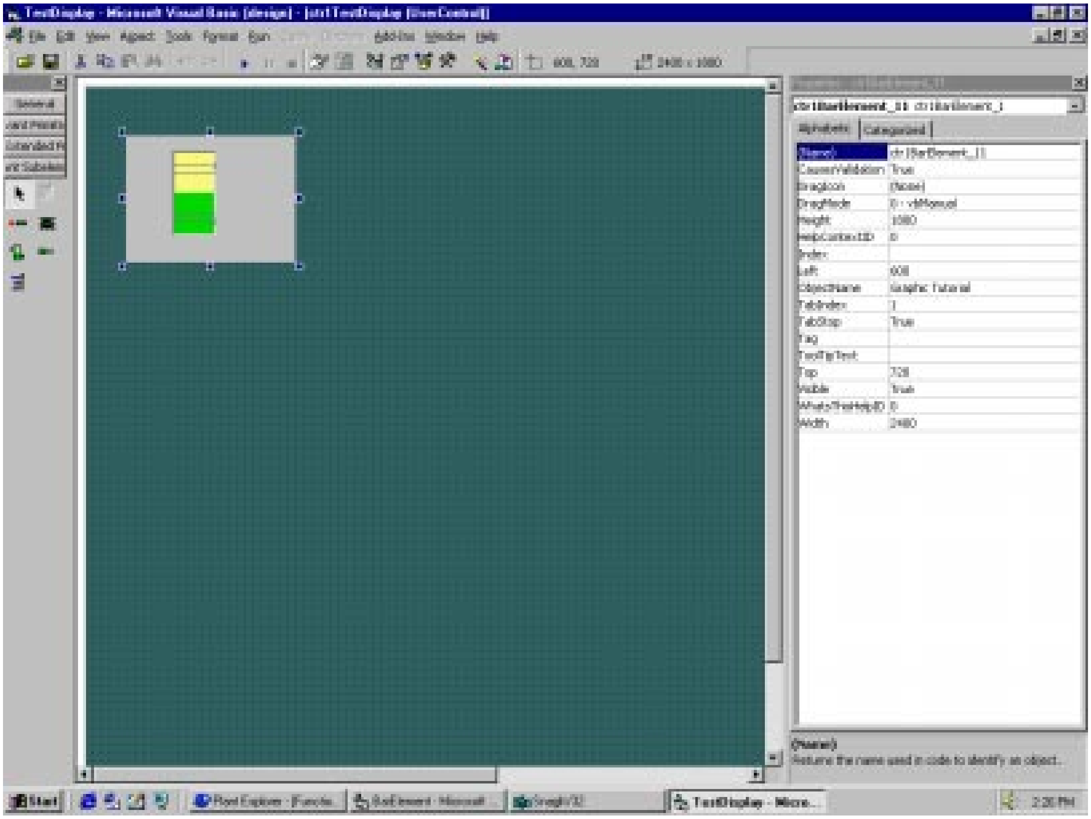


Figure 8-9. Adding an element to a graphic display

39. Enter the Object Name. In our example, we have used the name of an AIS system object.
40. Select the new element, right-click and select Copy from the context menu.

41. Move the cursor to the workspace, right-click and select Paste from the context menu.
42. Click **Yes** in the presented dialog box.

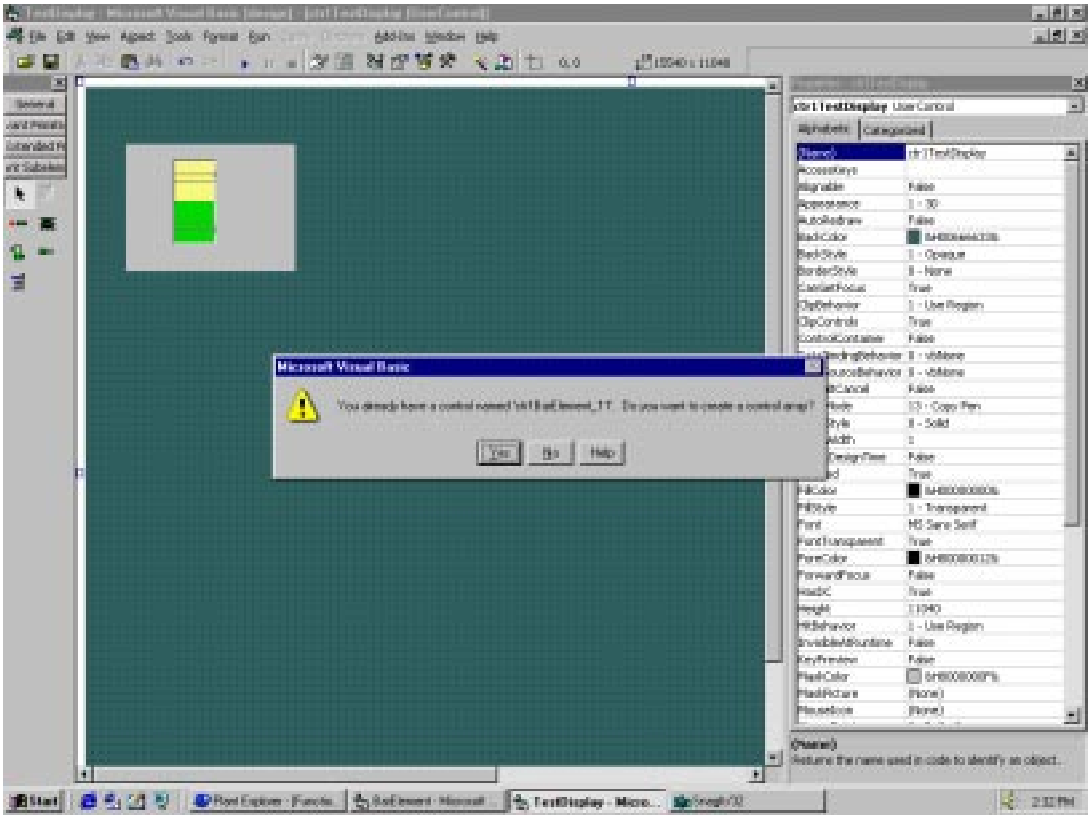


Figure 8-10. Adding an element to a graphic display

43. Select the copy and enter an Object Name. Use the name of another AIS system object.

44. Save and deploy the graphic display.
45. Exit Advant Graphics Builder.
46. Double-click on the aspect **Test Display** to call up the display.
47. Click on the first Bar Element in the display. The process dialog values are now directly controlled and changed by the value in the ABB Object **AIS1_LS1**.

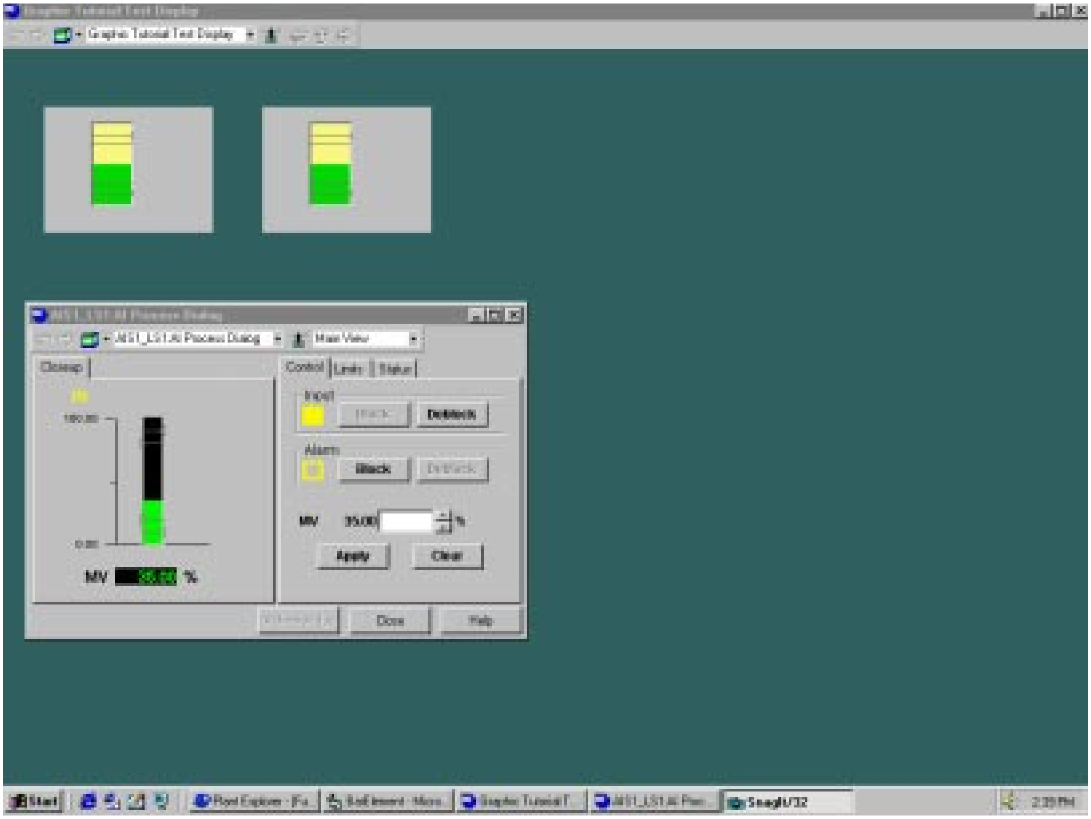


Figure 8-11. Controlling the process dialog values (one process dialog)

48. Click on the second Bar Element in the display. Note that the two otherwise similar bar elements presents different values.

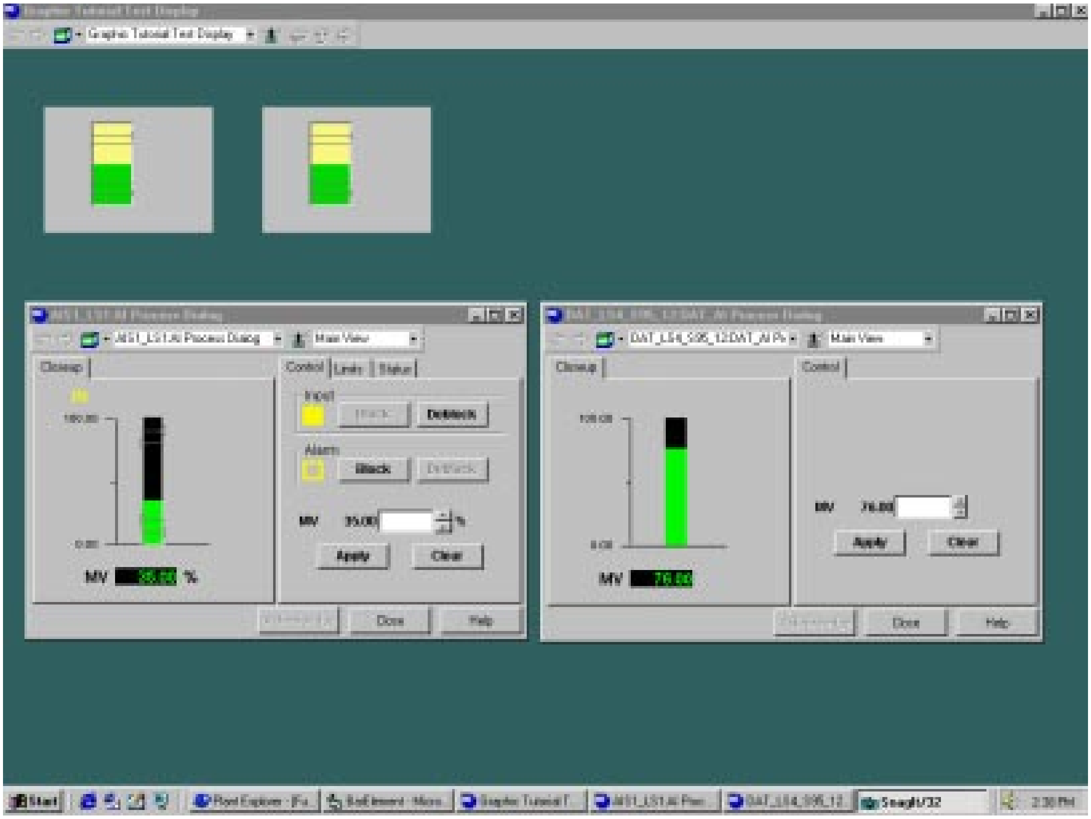


Figure 8-12. Controlling the process dialog values (two process dialogs)

8.10 Creating a Graphic Element and Using Visual Basic Code to Animate It

To create a graphic element and using Visual Basic code to animate it:

1. Create a new graphic aspect (a graphic element) in the ABB Object **Graphic Tutorial** in the Functional Structure, and call it **ValveElement_1**.
2. Right-click **ValveElement_1** and choose **Edit** from the context menu.
3. Select both the primitives and subelements on page three in the wizard, and click **Finish**.
4. Click on the **Advant Primitives** button in the toolbox.
5. Select the **AdvantPolygon**.
6. Create the valve element.

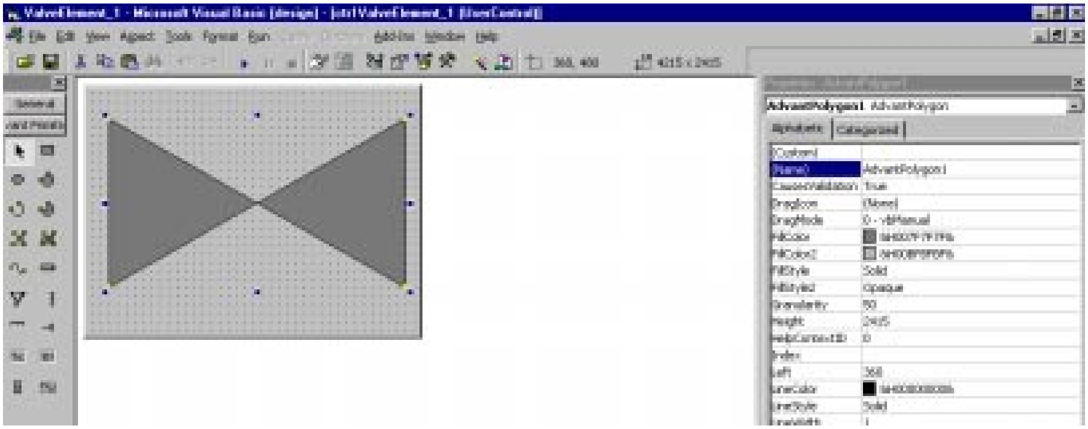


Figure 8-13. Creating a valve element

We would like to have colors that are selectable when we implement the element, so, we will need to add two members, one for color on, and one for color off.

7. Select **Add Members** in the **Tools** Menu.
8. Add the **ColorOn** and **ColorOff** properties and give them the Data Type of Color and select **OK**.

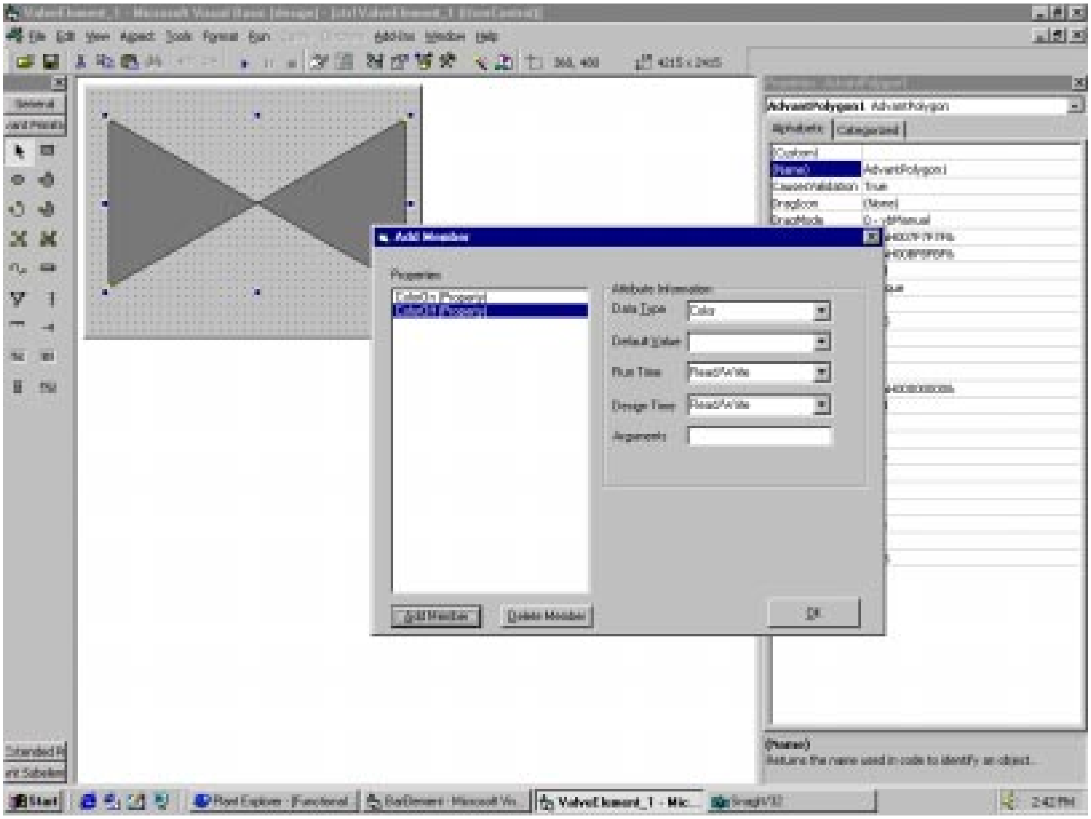
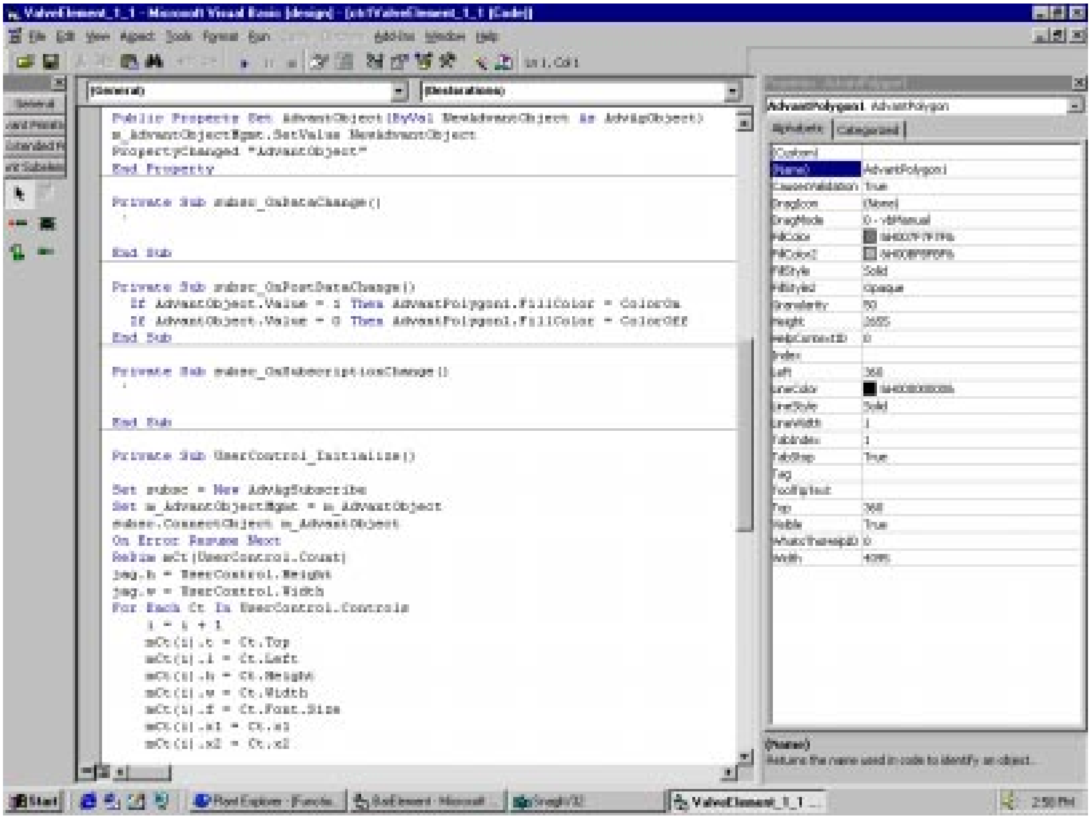


Figure 8-14. Adding members to the valve element

9. Double-click on the AdvantPolygon in the workspace to bring up the code window.
10. Select subsc_OnPostDataChange() in the Procedure List box and a template for the event procedure will be displayed in the code window.
11. The first set is to have the color of the AdvantPolygon to change based upon the value of the AdvantObject value property. To achieve this, enter the code as shown below in the subsc_OnPostDataChange().

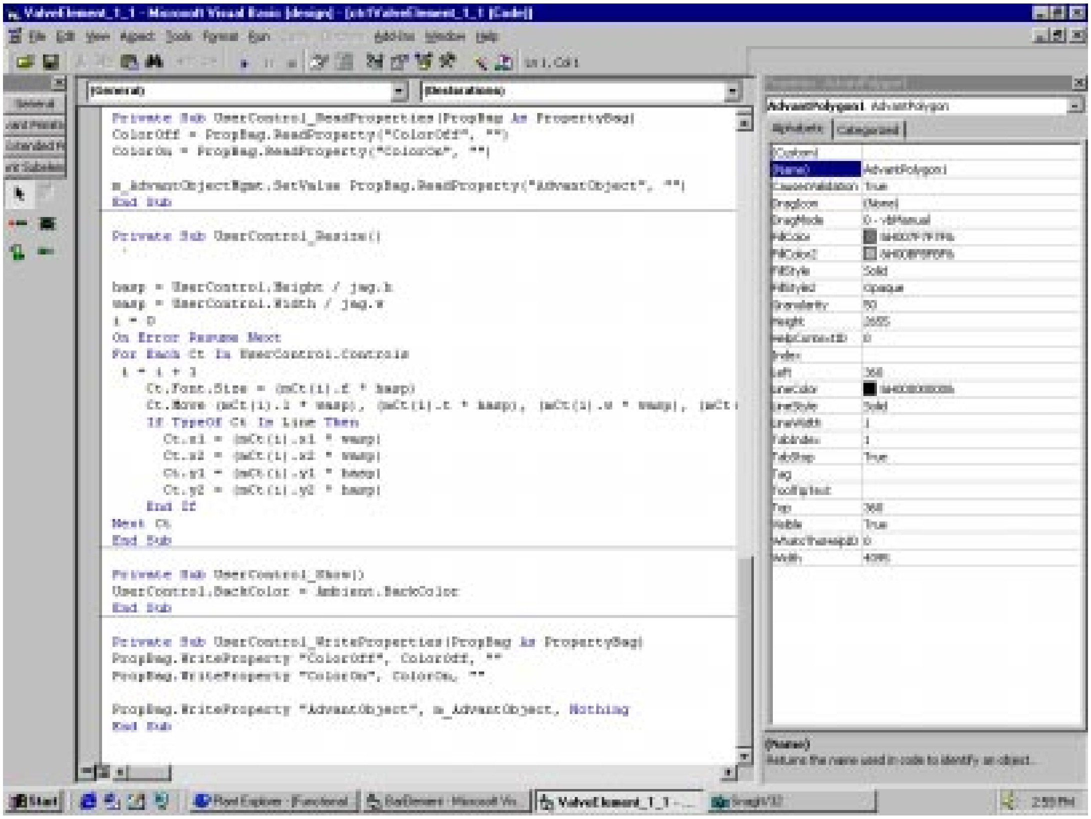


ValveElement_1_1 - Microsoft Visual Basic [design] - [ctrValveElement_1_1 (Code)]
File
Edit
View
Project
Format
Debug
Run
Add-Ins
Window
Help
Ln 1, Col 1
(General)
(Declarations)
AdvantPolygon1 AdvantPolygon
Alphabetic
Categorized
(Name)
Returns the name used in code to identify an object.
Start
ValveElement_1_1 ...
2:50 PM

```
Public Property Set AdvantObject(ByVal NewAdvantObject As AdvAgObject)
m_AdvantObjectMgmt.SetValue NewAdvantObject
PropertyChanged "AdvantObject"
End Property

Private Sub subsc_OnDataChange()
 '

End Sub

Private Sub subsc_OnPostDataChange()
  If AdvantObject.Value = 1 Then AdvantPolygon1.FillColor = ColorOn
  If AdvantObject.Value = 0 Then AdvantPolygon1.FillColor = ColorOff
End Sub

Private Sub subsc_OnSubscriptionChange()
 '

End Sub

Private Sub UserControl_Initialize()

Set subsc = New AdvAgSubscribe
Set m_AdvantObjectMgmt = m_AdvantObject
subsc.ConnectObject m_AdvantObject
On Error Resume Next
ReDim mCt(UserControl.Count)
img.h = UserControl.Height
img.w = UserControl.Width
For Each Ct In UserControl.Controls
    i = i + 1
    mCt(i).t = Ct.Top
    mCt(i).l = Ct.Left
    mCt(i).h = Ct.Height
    mCt(i).w = Ct.Width
    mCt(i).f = Ct.Font.Size
    mCt(i).x1 = Ct.x1
    mCt(i).x2 = Ct.x2
```

Figure 8-15. Entering code about the element’s color at value changes

12. Select Show in the Procedure List box and a template for the event procedure will be displayed in the code window.
13. Next, we would like to have the background color, the same as the workspace color. To achieve this, enter the code as shown below in the UserControl_Show() subroutine.



```
Private Sub UserControl_ReadProperties(PropBag As PropertyBag)
ColorOff = PropBag.ReadProperty("ColorOff", "")
ColorOn = PropBag.ReadProperty("ColorOn", "")

m_AdvantObjectMgmt.SetValue PropBag.ReadProperty("AdvantObject", "")
End Sub

Private Sub UserControl_Resize()
 '

hamp = UserControl.Height / jag.h
wamp = UserControl.Width / jag.w
i = 0
On Error Resume Next
For Each Ct In UserControl.Controls
 i = i + 1
    Ct.Font.Size = (mCt(i).f * hamp)
    Ct.Move (mCt(i).l * wamp), (mCt(i).t * hamp), (mCt(i).w * wamp), (mCt(
    If TypeOf Ct Is Line Then
      Ct.x1 = (mCt(i).x1 * wamp)
      Ct.x2 = (mCt(i).x2 * wamp)
      Ct.y1 = (mCt(i).y1 * hamp)
      Ct.y2 = (mCt(i).y2 * hamp)
    End If
Next Ct
End Sub

Private Sub UserControl_Show()
UserControl.BackColor = Ambient.BackColor
End Sub

Private Sub UserControl_WriteProperties(PropBag As PropertyBag)
PropBag.WriteProperty "ColorOff", ColorOff, ""
PropBag.WriteProperty "ColorOn", ColorOn, ""

PropBag.WriteProperty "AdvantObject", m_AdvantObject, Nothing
End Sub
```

Figure 8-16. Entering code about the element's background color

14. Save and deploy this element.

15. Create a new display and add the valve element **ValveElement_1**.
16. We can see that the ColorOn and ColorOff properties are now selectable when building the display. Select colors for the ColorOn and ColorOff properties.

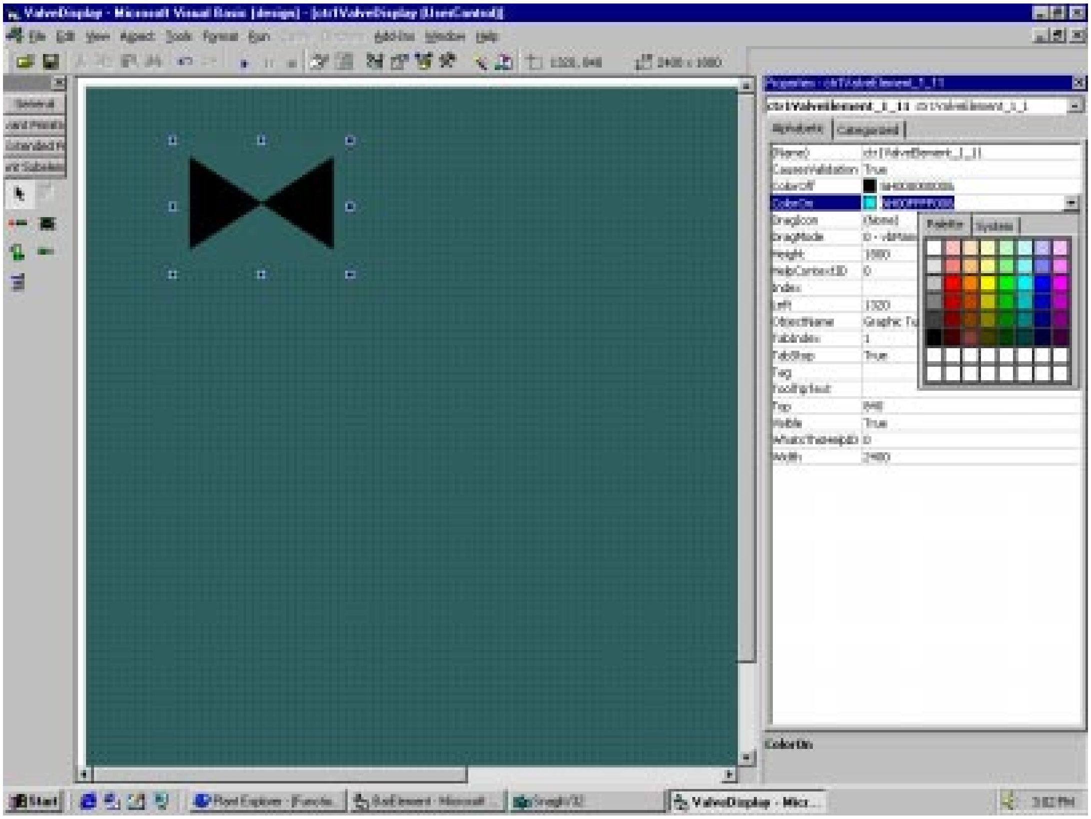
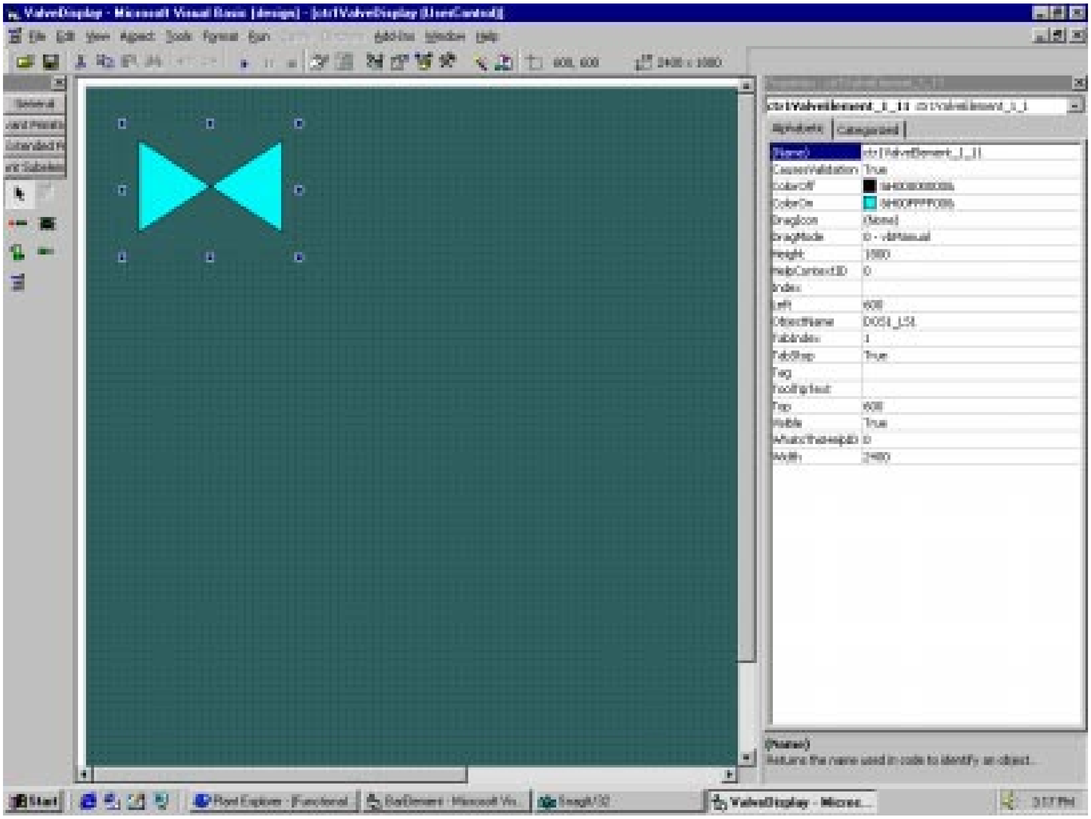


Figure 8-17. Selecting colors for the ColorOn and ColorOff properties

17. Enter a valid object name for the ObjectName property, then save and deploy the display. The figure below shows an example.



ValveDisplay - Microsoft Visual Basic [design] - [ctr1ValveDisplay (UserControl)]

(Name)	ctr1ValveElement_1_1
CausesValidation	True
ColorOff	&H00000000&
ColorOn	&H00FFFF00&
DragIcon	(None)
DragMode	0 - vbManual
Height	1800
HelpContextID	0
Index	
Left	600
ObjectName	DOS1_LS1
TabIndex	1
TabStop	True
Tag	
ToolTipText	
Top	600
Visible	True
WhatsThisHelpID	0
Width	2400

(Name)
Returns the name used in code to identify an object.

Figure 8-18. An animated graphic element in a graphic display (in Advant Graphics Builder)

18. Exit Advant Graphics Builder and call up the display by double-clicking on the display name in the aspect area.

19. Now when you click on the valve element in the display, you activate the process dialog for the object to which you have connected the element.

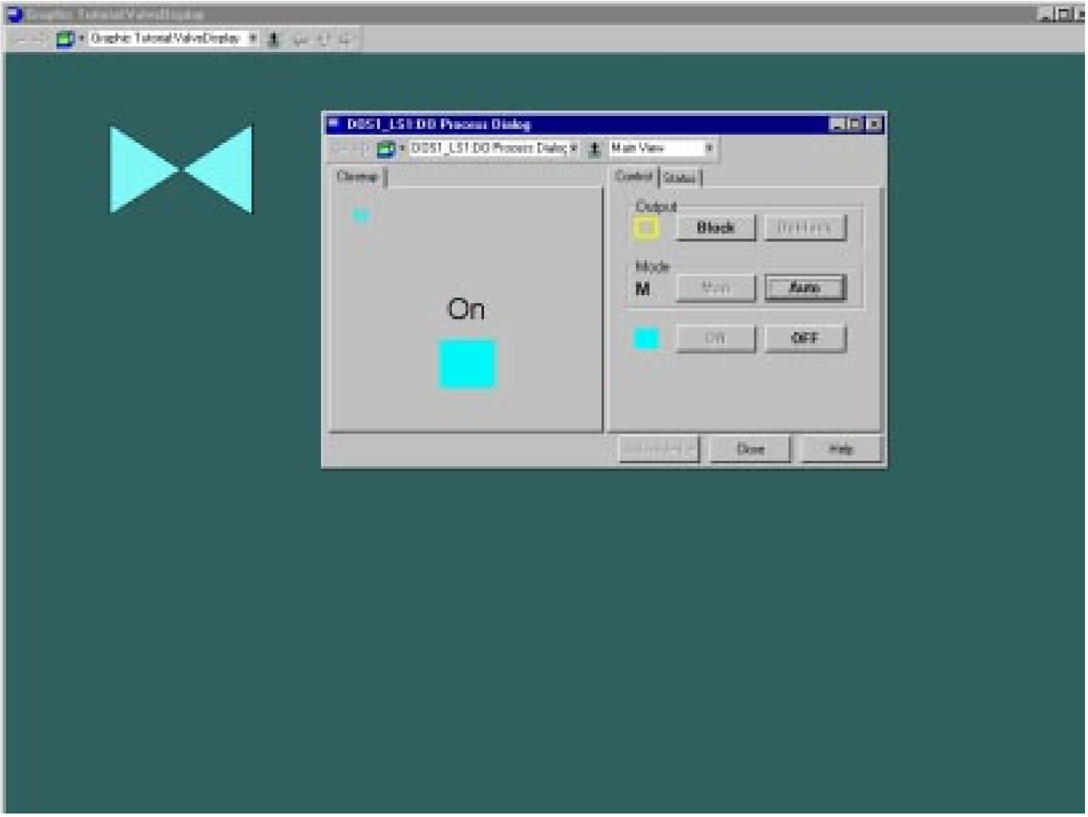

Graphic Tutorial ValveDisplay
DDS1_LS1:DO Process Dialog
DDS1_LS1:DO Process Dialog
Main View
Closeup
Control
Status
Output
Block
Mode
M
Auto
On
ON
OFF
Close
Help

Figure 8-19. A graphic display with an animated graphic element and its process dialog

Appendix A Basic Building Blocks Provided by ABB

A.1 Introduction

You build graphics by using the building blocks primitive elements, subelements, and graphic elements.

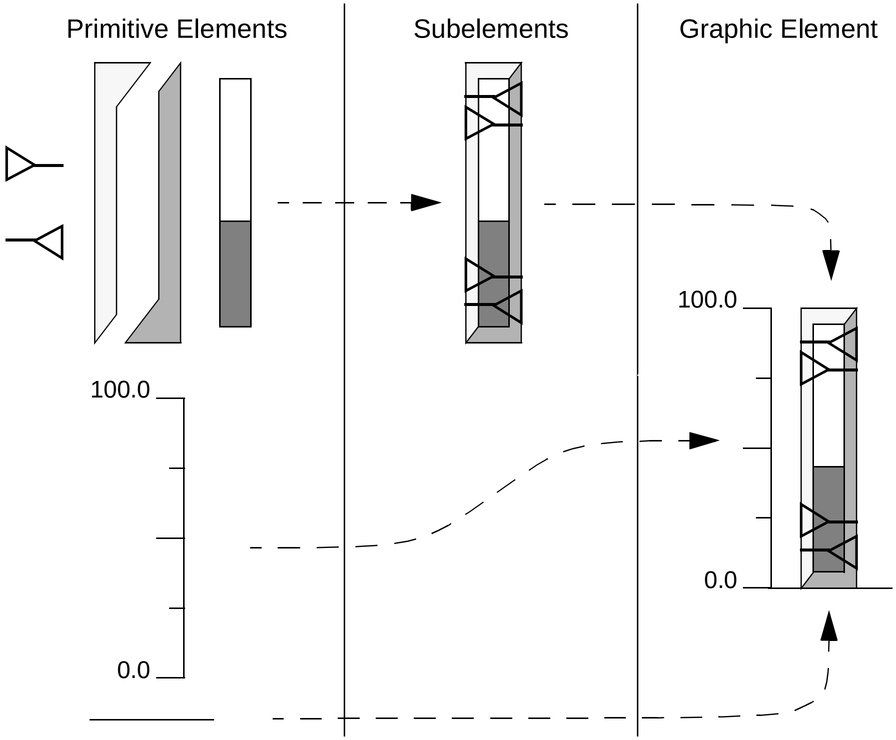


Figure A-1. Example of how a Graphic Element can be built up

There are no clear borderlines between the three individual element types, but the following definitions are used in this document.

- **Primitive element**

  A simple graphic element, for example a rectangle or a polyline. Normally a part of a graphic display. A graphical object typically built up from several subelements and/or primitive elements.

  ABB provides some basic primitive elements (Examples: AdvantArc, AdvantBar, AdvantChord), while others may be built by you or purchased from other sources in the form of graphic libraries. Primitive elements are graphic components (ActiveX controls) that are installed into system libraries.

  All elements that you use which haven't been built in Advant Graphics Builder are treated like primitive elements.

- **Subelement**

  A graphic element composed of primitive elements and suitable for data type presentation. Used to build graphic elements in Advant Graphics Builder.

  ABB provides some basic subelements (AdvantBar, AdvantValueField, AdvantObjectTrim and AdvantStatusLine), while others may be built by you or purchased from other sources in the form of graphic libraries. Examples: bar with limits, numeric value.

- **Graphic element**

  A graphic aspect that has a reference to an ABB Object. Normally a part of a graphic display. A graphic element typically built up from several subelements and/or primitive elements. ABB provides some graphic elements (see *Advant Connectivity Library for Advant Controller 100 Series Reference Manual*), while others may be built by you or purchased from other sources in the form of graphic libraries.

A.2 Advant Primitives (Primitive Elements)

Primitive elements are the smallest building blocks in Advant Graphics Builder. They are normally a part of a graphic display. You can find the basic primitive elements provided by ABB in the toolbox (see Section 3.10.2, Toolbox Window).

A.2.1 AdvantArc

An AdvantArc is a simple primitive element displaying a part of an ellipse.

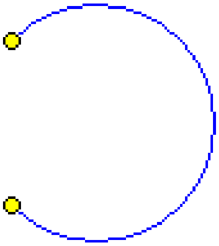


Figure A-2. AdvantArc

After creation the bounding rectangle can be manipulated to change size and position of the AdvantArc.

The properties of the primitive element AdvantArc are:

- Line Properties
- Angle Properties.

A.2.2 AdvantBar

An AdvantBar has the ability to show a vertically or a horizontally placed bargraph symbol.

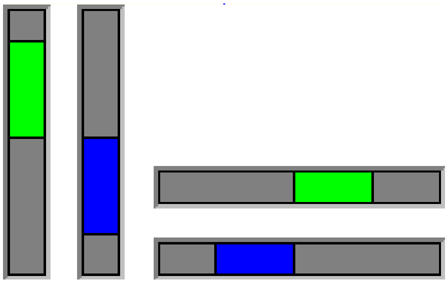


Figure A-3. AdvantBar

After creation the bounding rectangle can be manipulated to change size and position of the AdvantBar.

Table A-1. Example of AdvantBar Properties

Name	Type	Default	Description
MinValue	FLOAT	-100	Value of the bottom or left part of the bar.
MaxValue	FLOAT	100	Value of the top or right part of the bar.
StartValue	FLOAT	0	Starting point of the bar.
Value	FLOAT	50	Value to control the bar.
ValueColorMax	OLE_COLOR	00FF00	Color of the filled value part of the bar then Value is in the interval StartValue-MaxValue.
ValueColorMin	OLE_COLOR	0000FF	Color of the filled value part of the bar then Value is in the interval StartValue-MinValue.
Orientation	ENUM	VERTICAL	The orientation of the bar. Can be VERTICAL or HORIZONTAL.

Other properties of the primitive element AdvantBar are:

- Line Properties
- Fill Properties
- 3D Frame Properties.

A.2.3 AdvantBezier

An AdvantBezier is like a polyline where the connecting lines are formed using the bezier algorithm instead of straight lines.

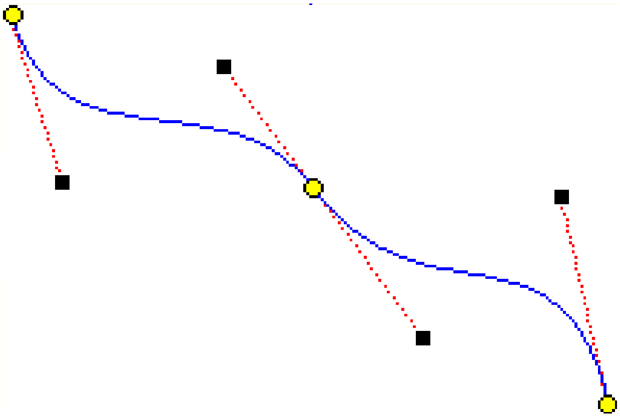


Figure A-4. AdvantBezier

Each point of a bezier curve is controlled by the point itself and two “gravity” points, which control the shape of the curve. The endpoints have only one “gravity” point.

After creation the bounding rectangle can be manipulated to change size and position of the AdvantBezier. See Section A.3.7, Point Properties for manipulating single points. Gravity points are moved in the same way as other points.

The property of the primitive element AdvantBezier is Line Properties.

A.2.4 AdvantChord

An AdvantChord draws a chord (a closed figure bounded by the intersection of an ellipse and a line segment).

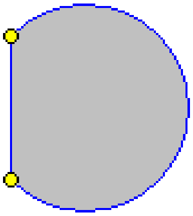


Figure A-5. AdvantChord

After creation the bounding rectangle can be manipulated to change size and position of the AdvantChord.

The properties of the primitive element AdvantChord are:

- Line Properties
- Fill Properties
- Angle Properties.

A.2.5 AdvantCone

The AdvantCone draws a cone consisting of four vertices connected by straight lines.

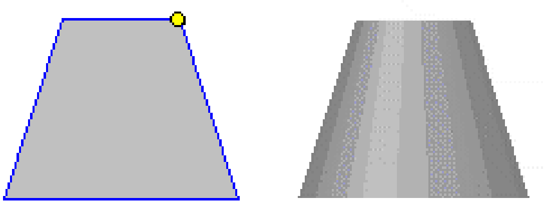


Figure A-6. AdvantCone

The cone can be any color, any line width, many different line styles, solid area fill, hatched area fill or shaded area fill.

After creation the bounding rectangle can be manipulated to change size and position of the AdvantCone.

Table A-2. Example of AdvantCone Properties

Name	Type	Default	Description
Direction	ENUM	UP	UP, DOWN, RIGHT or LEFT.
ConStrength	INTEGER	50	Size of the cone top in percent of full width. If set to 100, the cone top is the same size as the cone bottom, and the cone is drawn as a pipe with flat ends.

When using any of the shaded fill styles, the fill style will follow the direction of the cone.

Other properties of the primitive element AdvantCone are:

- Line Properties
- Fill Properties

A.2.6 AdvantEllipse

An AdvantEllipse can draw an ellipse with a frame and area fill.

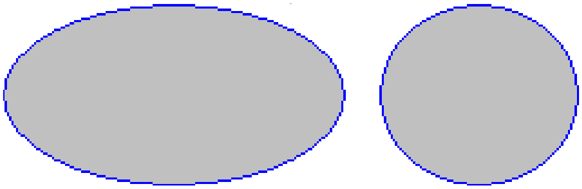


Figure A-7. AdvantEllipse

After creation the bounding rectangle can be manipulated to change size and position of the AdvantEllipse.

The properties of the primitive element AdvantEllipse are:

- Line Properties
- Fill Properties

A.2.7 AdvantHscale

An AdvantHscale has the ability to show a horizontally placed scale. The scale can contain labels as well as minor and major tics.

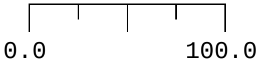


Figure A-8. AdvantHscale

After creation the bounding rectangle can be manipulated to change size and position of the AdvantHscale.

Table A-3. Example of AdvantHscale Properties

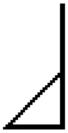
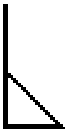
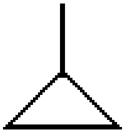
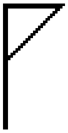
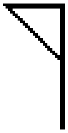
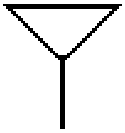
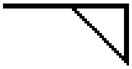
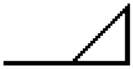
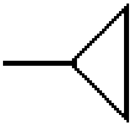
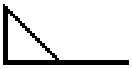
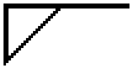
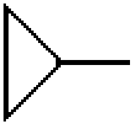
Name	Type	Default	Description
LabelFont	FONT	Default font from the container	Font used as labeling the scale.
LabelFormat	BSTR	#,#	Format of the scale labels in Excel format.
MajorTics	INTEGER	3	Number of major ticks.
MajorStrength	INTEGER	10	The length of the major ticks.
MinorTics	INTEGER	1	Number of minor ticks. The number is the amount of minor tics **between** two major tics.
MinorStrength	INTEGER	5	The length of the minor ticks.
LabelPosition	ENUM	DOWN	The position of the label and ticks. Can be UP or DOWN.
StartValue	FLOAT	0	The Start Value (Lowest point on the scale).
EndValue	FLOAT	100	The End Value (Highest point on the scale).

Another property of the primitive element AdvantHscale is Line Properties.

A.2.8 AdvantLimit

An AdvantLimit has the ability to show a vertically or horizontally placed limit symbol.

Table A-4. AdvantLimit

Examples	**LEFT**	**RIGHT**	**FULL**
UP			
DOWN			
LEFT			
RIGHT			

After creation the bounding rectangle can be manipulated to change size and position of the AdvantLimit.

Table A-5. Example of AdvantLimit Properties

Name	Type	Default	Description
Shape	ENUM	FULL	Shape of the filled part of the Limit. Can be LEFT, RIGHT or FULL.
Direction	ENUM	RIGHT	The direction of the Limit. Can be UP, DOWN, LEFT or RIGHT.

Other properties of the primitive element AdvantLimit are:

- Line Properties
- Fill Properties

A.2.9 AdvantNumeric

An AdvantNumeric has ability to show a number as a formatted text.

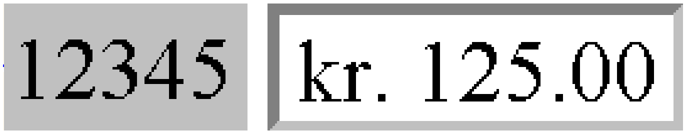


Figure A-9. AdvantNumeric

The Comment/Tooltip is used in a special way in the primitive element Numeric. If the formatted BSTR extends the available draw area, the font will be changed to a strikethrough font and the tooltip presented will be the full BSTR. If the formatted BSTR can be drawn in the draw area, the Comment is usually presented as a tooltip.

After creation the bounding rectangle can be manipulated to change size and position of the AdvantNumeric.

Table A-6. Example of AdvantNumeric Properties

Name	Type	Default	Description
Value	FLOAT	0.0	The value to display.
Format	BSTR	#,#	Format of the value in Excel format.
TextColor	OLE_COLOR	000000	Color of the text.
Font	FONT	Default font from the container	Font used in the text.
Halign	ENUM	CENTER	Horizontal alignment of the text inside the frame. Can be LEFT, CENTER or RIGHT.
Valign	ENUM	MIDDLE	Vertical alignment of the text inside the frame. Can be TOP, MIDDLE or BOTTOM.
StrikeThrough	BOOL	TRUE	Enable or disable the strike through facility, when the field is exhausted.

Other properties of the primitive element AdvantNumeric are:

- Fill Properties
- 3D Frame Properties.

A.2.10 AdvantObjectTrim

An AdvantObjectTrim has the ability to show up to four trim curves. The trim curve shows the most current 120 samples of the input, the Curve*N*Value property. The sample time is determined by the SampleTime property.

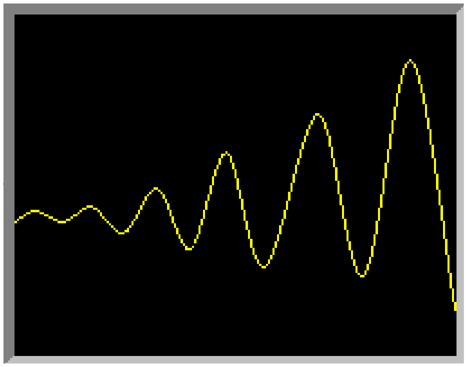


Figure A-10. AdvantObjectTrim

The trim curve is drawn in the color set in the Curve*N*Color property. The background of the trim area is determined by the Fill properties.

After creation the bounding rectangle can be manipulated to change size and position of the AdvantObjectTrim.

Table A-7. Example of AdvantObjectTrim Properties

Name	Type	Default	Description
SampleTime	INTEGER	1	Sample time in seconds.
GridVisible	BOOL	TRUE	Visibility of the grid.
GridColor	OLE_COLOR		Color of the grid.
GridNumberV	FLOAT	50	Number of vertical grid lines.
GridNumberH	FLOAT	0.5	Number of horizontal grid lines.
Curve*N*Visible	BOOL	TRUE	Visibility of the *N*:th curve.
Curve*N*Color	OLE_COLOR		Color of the *N*:th trend curve.
Curve*N*MinValue	FLOAT	-100	Minimum value of the trend area for the *N*:th curve.
Curve*N*MaxValue	FLOAT	100	Maximum value of the trend area for the *N*:th curve.
Curve*N*Value	FLOAT	0	Current value (input) for the *N*:th curve.

Other properties of the primitive element AdvantObjectTrim are:

- Fill Properties
- 3D Frame Properties.

A.2.11 AdvantPie

The AdvantPie draws a pie slice (a closed figure bounded by the intersection of an ellipse and line segments towards the center of the ellipse).

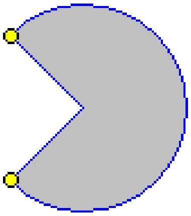


Figure A-11. AdvantPie

After creation the bounding rectangle can be manipulated to change size and position of the AdvantPie.

The properties of the primitive element AdvantPie are:

- Line Properties
- Fill Properties
- Angle Properties.

A.2.12 AdvantPipe

The AdvantPipe draws a pipe with area fill.

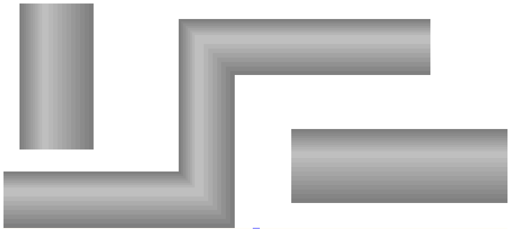


Figure A-12. AdvantPipe

The pipe can be any color, any line width, a number of different line styles, solid area fill, hatched area fill and shaded area fill. The pipe can be cut 45 degrees in one or both ends. AdvantPipe elements can be combined as shown above.

The pipe can be any color, any different line styles, solid area fill, hatched area fill or shaded area fill.

After creation the bounding rectangle can be manipulated to change size and position of the AdvantPipe.

Table A-8. Example of AdvantPipe Properties

Name	Type	Default	Description
Orientation	ENUM	VERTICAL	Orientation of the pipe. Can be VERTICAL or HORIZONTAL.
TopLeftEnd	ENUM	FLAT	Top or left end shape (depending of the direction) of the pipe. Can be FLAT, LEFT or RIGHT.
BottomRightEnd	ENUM	FLAT	Bottom or right end shape (depending of the direction) of the pipe. Can be FLAT, LEFT or RIGHT.

When using any of the shaded fill styles, the fill style will follow the orientation of the pipe.

Other properties of the primitive element AdvantPipe are:

- Line Properties
- Fill Properties

A.2.13 AdvantPolygon

An AdvantPolygon is a closed figure bounded by a line through a number of points.

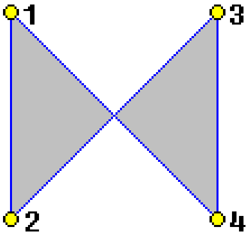


Figure A-13. AdvantPolygon

After creation the bounding rectangle can be manipulated to change size and position of the AdvantPolygon. See Section A.3.7, Point Properties for manipulation of single points.

The properties of the primitive element AdvantPolygon are:

- Line Properties
- Fill Properties
- Smooth

A.2.14 AdvantPolyline

An AdvantPolyline draws a line between a number of points.

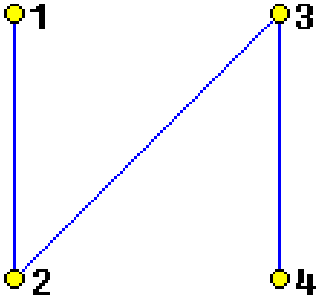


Figure A-14. AdvantPolyline

After creation the bounding rectangle can be manipulated to change size and position of the AdvantPolyline. See Section A.3.7, Point Properties for manipulating single points.

The properties of the primitive element AdvantPolyline are:

- Line Properties
- Smooth

A.2.15 AdvantRectangle

An AdvantRectangle can draw a rectangle with a frame and an area fill.

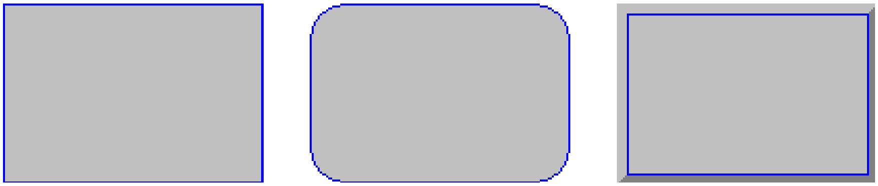


Figure A-15. AdvantRectangle

The frame can be any color, any line width, a number of different line styles, solid area fill, hatched area fill and shaded area fill. The corners of the rectangle can be sharp (90 degrees) or rounded.

After creation the bounding rectangle can be manipulated to change size and position of the AdvantRectangle.

Table A-9. Example of AdvantRectangle Properties

Name	Type	Default	Description
Round	INTEGER	0	Percent of rounding of the corners. 0 means a rectangle with no rounding, 100 means fully rounded.

Other properties of the primitive element AdvantRectangle are:

- Line Properties
- Fill Properties
- 3D Frame Properties.

A.2.16 AdvantText

An AdvantText has the ability to show a static text.

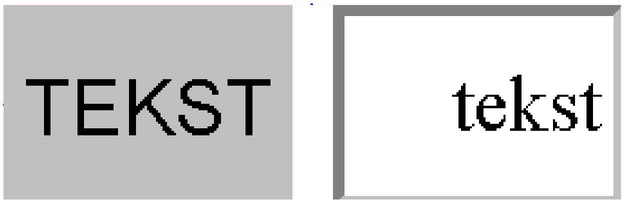


Figure A-16. AdvantText

The Comment/Tooltip is used in a special way in the primitive element AdvantText. If the Text BSTR extends the available draw area, the font will be changed to a strikethrough font and the tooltip presented will be the full text. If the Text BSTR can be drawn in the draw area, the Comment is usually presented as a tooltip.

After creation the bounding rectangle can be manipulated to change size and position of the AdvantText.

Table A-10. Example of AdvantText Properties

Name	Type	Default	Description
Text	BSTR	““ (empty)	The text to display.
TextColor	OLE_COLOR	000000	Color of the text.
Font	FONT	Default font from the container	Font used in the text.

Table A-10. Example of AdvantText Properties (Continued)

Name	Type	Default	Description
Halign	ENUM	CENTER	Horizontal alignment of the text inside the frame. Can be LEFT, CENTER or RIGHT.
Valign	ENUM	MIDDLE	Vertical alignment of the text inside the frame. Can be TOP, MIDDLE or BOTTOM.
StrikeThrough	BOOL	TRUE	Enable or disable the strike through facility, when the field is exhausted.

Other properties of the primitive element AdvantText are:

- Fill Properties
- 3D Frame Properties.

A.2.17 AdvantVscale

An AdvantVscale has the ability to show a vertically placed scale. The scale can contain labels as well as minor and major tics.

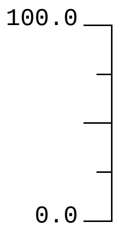


Figure A-17. AdvantVscale

After creation the bounding rectangle can be manipulated to change size and position of the AdvantVscale.

Table A-11. Example of AdvantVscale Properties

Name	Type	Default	Description
LabelFont	FONT	Default font from the container	Font used for labeling the scale.
LabelFormat	BSTR	#,#	Format of the scale labels in Excel format.
MajorTics	INTEGER	3	Number of major ticks.
MajorStrength	INTEGER	10	The length of the major ticks.
MinorTics	INTEGER	1	Number of minor ticks. The number is the amount of minor tics **between** two major tics.
MinorStrength	INTEGER	5	The length of the minor ticks.
LabelPosition	ENUM	LEFT	The position of the label and ticks. Can be LEFT or RIGHT.
StartValue	FLOAT	0	The Start Value (Lowest point on the scale).
EndValue	FLOAT	100	The End Value (Highest point on the scale).

Another property of the primitive element AdvantVscale is Line Properties.

A.2.18 Aspect Link

The Aspect link can be placed in a graphic display and used to open another aspect, either in an overlapping window or by replacing the current aspect. Note that it will always open an overlap from the preview area in the plant explorer, since the plant explorer does not allow replace.

The Aspect link will not work if placed in a graphic element placed in a graphic display. It must be placed at the top level.

The Aspect link is based on a standard command button.

Figure A-18. Aspect Link

Table A-12. Aspect Link Properties

Name	Type	Default	Description
Appearance	Integer	1	0 = Flat, 1 = 3D (See Command Button)
Aspect	String	-	The aspect this aspect link will open when pressed, in human-readable form.
AspectRef	IAfwAspectRef	-	The aspect this aspect link will open when pressed.

Table A-12. Aspect Link Properties (Continued)

Name	Type	Default	Description
BackColor	OLE_COLOR	Default background	The Background color of the aspect link. (See Command Button)
Caption	String	Aspect Link	The text displayed by the aspect link. (See Command Button)
DisabledPicture	Picture	-	The picture displayed by the aspect link when enabled is false. (See Command Button)
DownPicture	Picture	-	The picture displayed by the aspect link when it is pressed. (See Command Button)
Enabled	Boolean	True	If the aspect link accepts user input or not. This is forced to False if the aspect link is not at the top level in a graphic display or element.
Font	Font	MS Sans Serif	The Font used to display the Caption property. (See Command Button)
ForeColor	OLE_COLOR	Black	Currently not used.
Overlap	Boolean	False	If the aspect should be opened in an overlapping window or replace the current aspect.
Picture	Picture	-	The picture displayed by the aspect link.

A.3 Primitive Element Properties

All primitive elements can be assigned properties. The properties decide how the graphic element will appear. You change the properties from the Properties window (see Section 3.13.1, Properties Window).

A.3.1 Line Properties

All lines drawn in the primitive elements are controlled by the Line Width, Line Color and Line Style.

Table A-13. Line Properties

Name	Type	Default	Description
LineWidth	INTEGER	1	Width in pixels of the drawn line. Valid values are any positive integer.
LineColor	COLOR	000000	Color of the line.
LineStyle	ENUM	SOLID	Style of the line. See Section A.3.1.1, Line Styles below.

Fill properties applies to the primitive elements AdvantArc, AdvantBar, AdvantBezier, AdvantChord, AdvantCone, AdvantEllipse, AdvantHscale, AdvantLimit, AdvantObjectTrim, AdvantPie, AdvantPipe, AdvantPolygon, AdvantPolyline, AdvantRectangle and AdvantVscale.

A.3.1.1 Line Styles

The LineStyle property specifies the style of the line.

Table A-14. Line Styles

Value	Enum	Sample
0	SOLID	────────────────
1	DASH	— — — — — — — — — — — — — —
2	DOT	....

Table A-14. Line Styles (Continued)

Value	Enum	Sample
3	DASHDOT	— - — - — - — - — - — - — - — - — - -
4	DASHDOTDOT	— - - — - - — - - — - - — - - — - - — - -

Remark: Solid lines can be any width. Any other style uses a line width of 1 no matter of the value in LineWidth.

A.3.2 Fill Properties

Numerous numbers of the primitive elements are area filled. The filling is controlled by FillColor, FillStyle and FillColor2. The FillStyle2 is effective when one of the hatched or shading patterns is used. The FillStyle property specifies how the area fill would appear.

Table A-15. Fill Properties

Name	Type	Default	Description
FillColor	COLOR	BFBFBF	Color of the area fill.
FillStyle	ENUM	SOLID	See Section A.3.2.1, Fill Styles below.
FillColor2	COLOR	7F7F7F	Second color of the rectangle if FillStyle is set to any of the hatched or shading methods.
FillStyle2	ENUM	OPAQUE	Mode of the background if any of the hatched patterns are used. Can be OPAQUE or TRANSPARENT.

Fill properties applies to the primitive elements AdvantChord, AdvantLimit, AdvantPie and AdvantPolygon.

A.3.2.1 Fill Styles

The FillStyle property specifies how the area fill would appear.

Table A-16. Fill Styles

Value	Enum	Sample
0	SOLID	
1	TRANSPARENT	
2	BDIAGONAL	▧
3	CROSS	▦
4	DIACROSS	▩
5	FDIAGONAL	▨

Table A-16. Fill Styles (Continued)

Value	Enum	Sample
6	HORIZONTAL	▤
7	VERTICAL	▥

A.3.3 Fill Properties with Shade

Numerous numbers of the primitive elements are area filled. The filling is controlled by FillColor, FillStyle and in case of shading also by FillColor2 and Granularity. The FillStyle2 is effective when one of the hatched patterns is used. The FillStyle property specifies how the area fill would appear.

Table A-17. Fill Properties

Name	Type	Default	Description
FillColor	COLOR	BFBFBF	Color of the area fill.
FillStyle	ENUM	SOLID	See Section A.3.3.1, Fill Styles below.
FillColor2	COLOR	7F7F7F	Second color of the rectangle if FillStyle is set to any of the hatched or shading methods.
FillStyle2	ENUM	OPAQUE	Mode of the background if any of the hatched patterns are used. Can be OPAQUE or TRANSPARENT.
Granularity	INTEGER	50	Granularity of the shading.

Fill properties with shade applies to the primitive elements AdvantBar, AdvantCone, AdvantEllipse, AdvantNumeric, AdvantPipe, AdvantRectangle, AdvantText and AdvantObjectTrim.

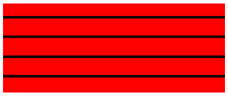
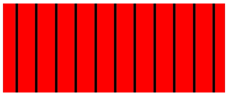
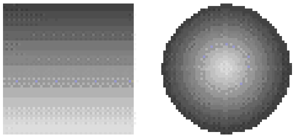
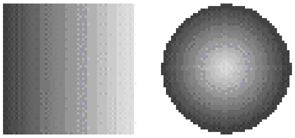
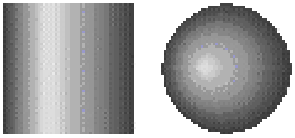
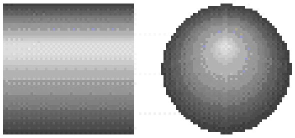
A.3.3.1 Fill Styles

The FillStyle property specifies how the area fill would appear.

Table A-18. Fill Styles

Value	Enum	Sample
0	SOLID	
1	TRANSPARENT	
2	BDIAGONAL	▧
3	CROSS	▦
4	DIACROSS	▩
5	FDIAGONAL	▨

Table A-18. Fill Styles (Continued)

Value	Enum	Sample
6	HORIZONTAL	
7	VERTICAL	
8	VERTICAL SHADE	
9	HORIZONTAL SHADE	
10	VERTICAL 3DSHADE	
11	HORIZONTAL 3DSHADE	

A.3.4 3D Frame Properties

All rectangle shaped primitive elements have the ability to show a 3D frame around the element. The 3D frame is controlled by FrameWidth, FrameColor, FrameColor2 and 3D-Effect.

Table A-19. 3D Frame Properties

Name	Type	Default	Description
FrameWidth	INTEGER	1	Width of the 3D frame.
FrameColor	COLOR	DFDFDF	Color of the left-top part of the 3D frame.
FrameColor2	COLOR	7F7F7F	Color of the right-bottom part of the 3D frame.
3D-Effect	ENUM	RAISED	3D effect of the frame. See Section A.3.4.1, 3D Effect below.

3D Frame properties applies to the primitive elements AdvantBar, AdvantNumeric, AdvantObjectTrim, AdvantRectangle and AdvantText.

A.3.4.1 3D Effect

The 3D effect of the frame can be RAISED or SUNKEN. If set to RAISED FrameColor is used for the left-top part. FrameColor2 is used for the bottom-right part. If set to SUNKEN the colors will be swapped.

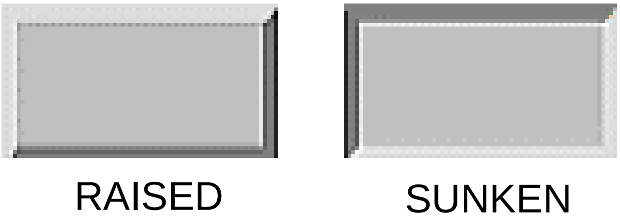


Figure A-19. 3D-Effect Property (Raised/Sunken)

A.3.5 Smooth

Smooth is defined in the range 0..100. When Smooth is set > 0, the function includes as many (visible) points as specified, between the original points. Then the bezier algorithm shapes each straight line into an arc instead.

Smooth applies to the primitive elements AdvantPolygon and AdvantPolyline.

A.3.6 Angle Properties

The primitive elements AdvantArc, AdvantChord and AdvantPie have round handles at the start and end points.

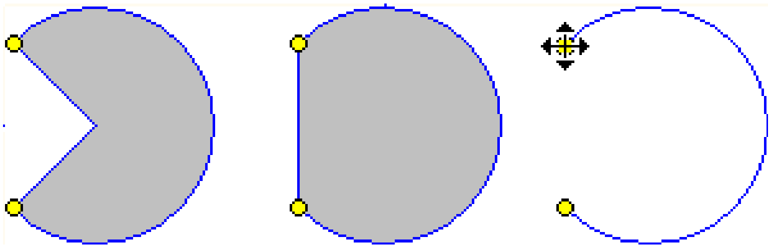


Figure A-20. Primitive Elements with Angle Properties

The angles can be changed either by holding down the right mouse button while dragging the handles (the cursor changes to AllsizeArrow when moved by a handle) or by entering the desired values in the StartAngle and StopAngle properties.

Table A-20. Angle Properties

Name	Type	Default	Description
StartAngle	INTEGER	225	Starting angle (0-359) of the arc. 0 is positioned at 3 o’clock.
StopAngle	INTEGER	135	Stop angle (0-359) of the arc. If StartAngle and StopAngle are equal a full ellipse is drawn.

Angle properties applies to the primitive elements AdvantArc, AdvantChord and AdvantPie.

A.3.7 Point Properties

Some of the primitive elements are defined by a number of connected points.

Point properties applies to the primitive elements AdvantBezier, AdvantPolygon and AdvantPolyline.

A.3.7.1 To Add a New Point

To add a new point:

1. Move the cursor to a line between two points. The cursor changes to a crosshair (+).
2. Press Shift + left mouse button. The new point is added.

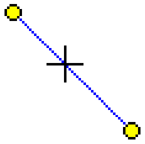


Figure A-21. Adding a New Point

A.3.7.2 To Delete a Point

To delete a point:

1. Move the cursor to the point. The cursor changes to AllsizeArrow.
2. Press Ctrl + left mouse button. The point is deleted.

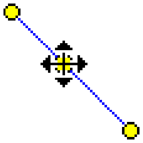


Figure A-22. Deleting a Point

A.3.7.3 To Move a Point

To move a point:

1. Move the cursor to the point. The cursor changes to AllsizeArrow.
2. Press left mouse button and drag. The point is moved.

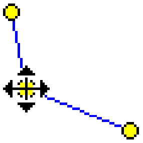


Figure A-23. Moving a Point

A.4 Advant Subelements

These are basic subelements developed by ABB. They are building blocks in Advant Graphics Builder. You can find the subelements provided by ABB in the toolbox (see Section 3.10.2, Toolbox Window).

A.4.1 AdvantBar

An AdvantBar has the ability to show a bar with two hatched limit areas or four limit markers.

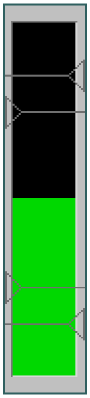


Figure A-24. AdvantBar

After creation the bounding rectangle can be manipulated to change size and position of the AdvantBar.

Table A-21. AdvantBar Properties

Property Name	Default	Remarks
Style	FourLimits	FourLimits or TwoLimits
Orientation	Vertical	Bar orientation, Vertical or Horizontal
Value	50	Value level

Table A-21. AdvantBar Properties (Continued)

Property Name	Default	Remarks
StartValue	0	Start position of value in the bar
RangeMax	100	Top value of the bar
RangeMin	0	Bottom value of the bar
ValueColor	Green	Color of bar and question mark
ValueColorNeg	Blue	Color of bar when value is below Start Value
BackColor	Black	Background color of the bar
Error	False	True: ErrorCaption is shown. The bar is hidden. False: The bar is shown
ErrorCaption	"?"	Indication for Error presentation
Font	Arial 12 Bold	Font used for ErrorCaption mark
HighLimit2	85	Alarm limit high marker placement
HighLimit2Fill	Transparent	Solid or Transparent
HighLimit2Color	Grey	Color of the limit and filling
HighLimit2Visible	True	Controls visibility, True or False
HighLimit1	75	Warning limit high marker placement
HighLimit1Fill	Transparent	Solid or Transparent
HighLimit1Color	Grey	Color of the limit and filling
HighLimit1Visible	True	Controls visibility, True or False
LowLimit1	25	Warning limit low marker placement
LowLimit1Fill	Transparent	Solid or Transparent
LowLimit1Color	Grey	Color of the limit and filling
LowLimit1Visible	True	Controls visibility, True or False
LowLimit2	15	Alarm limit low marker placement

Table A-21. AdvantBar Properties (Continued)

Property Name	Default	Remarks
LowLimit2Fill	Transparent	Solid or Transparent
LowLimit2Color	Grey	Color of the limit and filling
LowLimit2Visible	True	Controls visibility, True or False
FrameStyle	Sunken	Raised or Sunken
FrameWidth	1	

A.4.1.1 Bar Styles

The subelement AdvantBar can have two hatched limit areas or four limit markers.

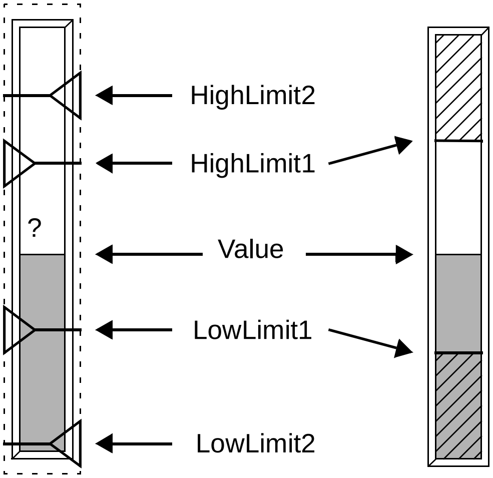


Figure A-25. Bar Styles

FourLimits

The height of the limits is 1/11 of the subelement height. The height of the bar area (bar without frame) is 10/11 of the element height. The vertical edges of the bar area is placed on the center of the limit headers. The FrameWidth cannot exceed 1/2 of the width of a limit header.

TwoLimits

The bar is resized to fit the subelement. The limits are sized to fit the bar area. The lower edge of the upper limit area is determined by HighLimit1. The upper edge of the lower limit area is determined by LowLimit1.

The properties below are set to the given value in accordance with the chosen Style.

Table A-22. Default Property Values for Bar Styles

Property Name	FourLimits	TwoLimits
Field1Visible	True	True
Orientation	Vertical	Vertical
Value	50	50
StartValue	0	0
RangeMax	100	100
RangeMin	0	0
ValueColor	Green	Green
ValueColorNeg	Blue	Blue
BackColor	Black	Black
Error	False	False
ErrorCaption	“?”	“?”
Font	Arial 10 Bold	Arial 10 Bold

Table A-22. Default Property Values for Bar Styles (Continued)

Property Name	**FourLimits**	**TwoLimits**
HighLimit2	85	
HighLimit2Fill	Transparent	
HighLimit2Color	Grey	
HighLimit2Visible	True	
HighLimit1	75	75
HighLimit1Fill	Transparent	Transparent
HighLimit1Color	Grey	Grey
HighLimit1Visible	True	True
LowLimit1	25	25
LowLimit1Fill	Transparent	Transparent
LowLimit1Color	Grey	Grey
LowLimit1Visible	True	True
LowLimit2	15	
LowLimit2Fill	Transparent	
LowLimit2Color	Grey	
LowLimit2Visible	True	
FrameStyle	Sunken	Sunken
FrameWidth	1	1

A.4.2 AdvantValueField

An AdvantValueField has the ability to show a value and a unit.

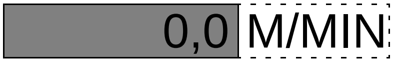


Figure A-26. AdvantValueField

The minimum height of the subelement is determined by the chosen font.

The width of UnitField is determined by the Unit property and the chosen font. When changing the unit, the width of the subelement is changed to keep width of the Value field unchanged. The width of Value field is changed by setting the width of the subelement directly.

Table A-23. AdvantValueField Properties

Property Name	Default	Remarks
Value	0	Double
ValueColor	Green	Color of value
ValueFormat	"**0.0**"	Format string
ValueFont	Arial, Bold, 12	Font for the value field
ValueBackColor	Black	Background color of value field
Error	False	True: 3 question marks in Value field False: The value is shown in Value field
Unit	"M/MIN"	String
UnitColor	Grey	Color of unit
UnitFont	Arial, Regular, 12	

Table A-23. AdvantValueField Properties (Continued)

Property Name	Default	Remarks
FrameStyle	Flat	Flat, Raised, Sunken, surrounds ValueField
FrameWidth	1	
FrameColor	Black	Only used for FrameStyle = Flat, otherwise Windows default grey frame colors

A.4.3 AdvantObjectTrim

An AdvantObjectTrim has the ability to show a trim curve with the most current 240 samples of the input, the Value property. The sample time is determined by the UpdateRate property. The default UpdateRate is set to 1 which makes the presentation area cover four minutes for this update rate.

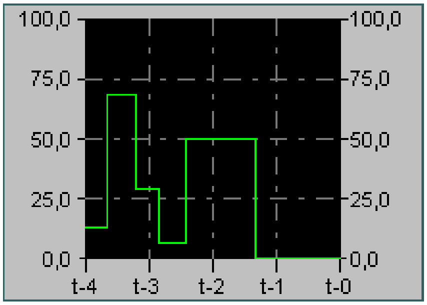


Figure A-27. AdvantObjectTrim

The width of the left and right scale and the height of the time labels are determined by the chosen font. Constituent controls are then resized and positioned to fill out the subelement and to fit ticks with grid. The number of time ticks and corresponding grid lines is set to 5.

Table A-24. AdvantObjectTrim Properties

Property Name	Default	Remarks
UpdateRate	0	Time in sec. between updates, "0" means update deactivated
Value	0	The input value (double)
ValueColor	Green	Color of the curve
BackColor	Black	Background color of value field
GridVisible	False	Visibility of the grid
RangeMax	100	Top value of the scale and curve
RangeMin	0	Bottom value of the scale and curve
ScaleColor	Grey	Color of the scale
ScaleFont	Arial, Regular, 8	Font of the scale labels
NumOfTics	5	Number of ticks for value

A.4.4 AdvantStatusLine

An AdvantStatusLine has the ability to show a status line in one of four styles.

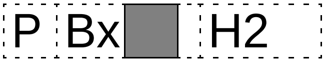


Figure A-28. AdvantStatusLine

The height of the subelement AdvantStatusLine is determined by the chosen font.

Table A-25. AdvantStatusLine Properties

Property Name	Default	Remarks
Style	Style0	Style0, Style1, Style2 or Style3
Field1Visible	True	Boolean
Field1Caption	"**P**"	String
Field1Color	Yellow	Color of caption
Field2Visible	True	Boolean
Field2Caption	"**Bx**"	String
Field2Color	Yellow	Color of caption
IndicatorVisible	True	Boolean
IndicatorCaption	""	String
IndicatorForeColor	Red	Color of caption
IndicatorBackColor	Red	Color of background
IndicatorFillStyle	Solid	Solid or Transparent
TextCaption	"**H2**"	String
TextColor	Grey	Color of caption
TextVisible	True	Boolean
Value	0	Double
ValueFormat	"**0.0**"	Format string
ValueColor	Green	Color of value
ValueVisible	True	Boolean
ValueBackColor	Black	Color of background

Table A-25. AdvantStatusLine Properties (Continued)

Property Name	Default	Remarks
ValueFillStyle	Solid	Solid or Transparent
Font	Arial Bold 12	Font used for all fields

A.4.4.1 Status Line Styles

The subelement AdvantStatusLine can have four styles with a different number of properties.

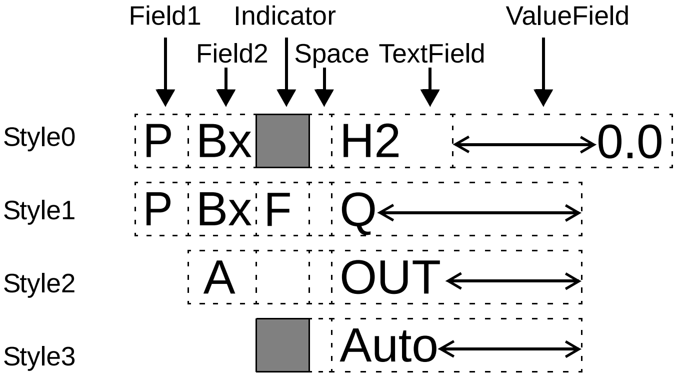


Figure A-29. Status Line Styles

Style0

The widths of Field1, Field2, Indicator and TextField are determined by the chosen font. Field1 and Indicator are square. The width of Field2 and TextField are set to fit the string “_Bx“, respectively “_H2“. The width of the space between Indicator and TextField is set to 120 twips. The minimum width of the subelement is the sum of widths of Field1, Field2, Indicator TextField and the space between Indicator and TextField. Additional horizontal space, if any, is allocated to the ValueField.

Style1

Like Style0, except that ValueField is set invisible (forced) and not used in the layout. The minimum width of the subelement is the sum of widths of Field1, Field2, Indicator and the space between Indicator and TextField. Additional horizontal space, if any, is allocated to the TextField.

Style2

Like Style1, except that Field1 is set invisible (forced) and not used in the layout. The minimum width of the subelement is the sum of widths of Field2, Indicator and the space between Indicator and TextField. Additional horizontal space, if any, is allocated to the TextField.

Style3

Like Style2, except that Field2 is set invisible (forced) and not used in the layout. The minimum width of the subelement is the width of Indicator and the space between Indicator and TextField. Additional horizontal space, if any, is allocated to the TextField.

The properties below are set to the given value in accordance with the chosen Style.

Table A-26. Default Property Values for Status Line Styles

Property Name	Style1	Style2	Style3	Style4
Field1Visible	True	True	False	False
Field1Caption	“**P**”	“**P**”		
Field1Color	Yellow	Yellow		
Field2Visible	True	True	True	False
Field2Caption	“**Bx**”	“**Bx**”	“**A**”	
Field2Color	Yellow	Yellow	Magenta	

Table A-26. Default Property Values for Status Line Styles (Continued)

Property Name	**Style1**	**Style2**	**Style3**	**Style4**
IndicatorVisible	True	True	False	True
IndicatorCaption	“”	“**F**”		“”
IndicatorForeColor	Red	Red		Red
IndicatorBackColor	Red			Green
IndicatorFillStyle	Solid	Transparent		Solid
TextVisible	True	True	True	True
TextCaption	“**H2**”	“**Q**”	**“OUT”**	“**Auto**”
TextColor	Grey	Grey	Yellow	Grey
ValueVisible	True	False	False	False
Value	0			
ValueFormat	“**0.0**”			
ValueColor	Green			
ValueBackColor	Black			
ValueFillStyle	Solid			
Font	Arial Bold 12	Arial Bold 12	Arial Bold 12	Arial Bold 12

Appendix B Expressions

B.1 Introduction

Expressions are used for performing two tasks:

- Specify subscriptions
- Specify a relationship between process data and data to be displayed.

The expression language is based on standard Visual Basic. However, some additions are made to easily set up a subscription.

To each property of the contained ActiveX control, either an expression or a constant value can be applied, see Figure B-1.

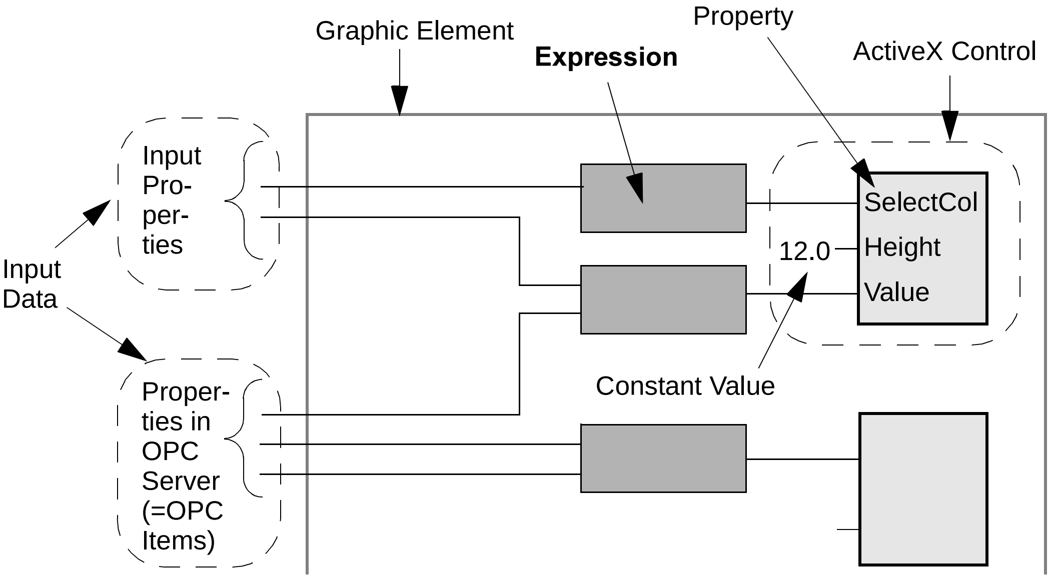


Figure B-1. Expressions are hooked up on ActiveX Control properties

Expressions are defined within graphic elements using the Expressions window and the Expression Builder window. An expression defines calculations on data.

Input data for an expression can be:

- Input properties of the graphic display.
- Properties of process objects in an OPC server (OPC items).

The execution of an expression takes place automatically when some input data to the expression changes. The execution takes place for following reasons:

- Initial when the graphic element is presented. The initial execution handles constant values applied to input properties of the graphic element.
- When values of an input property changes for any reason (not automatically, except for the AdvantObject property).
- When data is delivered from the OPC server.

B.2 ABB Expressions

Expressions are assigned to properties of contained ActiveX controls using the Expressions window or the Expression Builder window (see Section 3.15.1, Assigning Expressions and Section 3.16.1, Expression Builder Window).

The ABB extensions of Visual Basic expressions is that "item references" can be used in an expression at any point where Visual Basic allows a variable to be used.

The following is a legal expression:

$'AI.117'.VALUE * 14.2

which means that the VALUE property of object AI.117 multiplied with 14.2 drives the property to which this expression is applied.

See Section B.2.2, ABB Expressions for a detailed description of the legal syntax for item references.

B.2.1 Object References

The structure in Figure B-2 is used in the examples in this and the following sections.

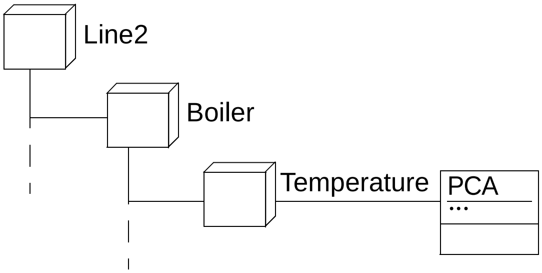


Figure B-2. Example of a structure

Input properties can be object references. These properties accept an ABB Object as value.

An object reference property (for example AdvantObject) can be initialized with a constant (lexical) value using the name server syntax, for example:

Line2/Boiler

All constant values are assigned in the Properties window.

The AdvantObject property is not visible in the Properties window. It is represented by the ObjectName property, which is the string representation of the AdvantObject property.

An object reference property (for example AdvantObject) can also be assigned an "object reference expression" using the syntax specified in Section B.2.2, ABB Expressions, for example:

AdvantObject.subPath("Boiler")

B.2.2 Syntax for Item References

The syntax for an item reference has the following general format:

ObjectRef.PropertyRef

where ObjectRef specifies an ABB Object and PropertyRef the name of the property of the subscribed object.

B.2.2.1 ObjectRef

The ObjectRef part largely resembles the general syntax supported by the Afw Name Server.

The following variations exists:

$'<Name server syntax>'

ObjectRefVar[.subPath("<Name server syntax>")]

where ObjectRefVar is an object reference variable.

Examples:

$'Line2/Boiler/Temperature'

$'Line2'.subPath("Boiler/Temperature")

AdvantObject.subPath("Boiler/Temperature")

The examples above shows how name paths and relative names can be specified. Additions for specifying structure transitions and name types, in accordance with the name server syntax, are supported as well.

An object reference variable typically is the AdvantObject input property. By being able to add a sub path to an input property, a graphic element can be designed to support a structure of objects, that is, a type solution.

B.2.2.2 PropertyRef

The PropertyRef part is the name of a property (item in OPC terminology) in relation to an aspect. If the property is omitted, and the receiving property of the ActiveX control is not an ABB Object property, the default property name is used.

If there are more than one aspect that has properties with the same name, the aspect must be specified using the syntax [aspect:property].

Examples:

AdvantObject.Value

$'Line2/Boiler/Temperature'.Value

B.3 Visual Basic Operators

Beneath the Edit box in the Expression Builder window is a row of buttons. The purpose of the buttons is to provide help with the syntax of expressions. The symbol or function name (that is, the operators) on the button is added to the expression in the Edit box when the button is pressed.

The buttons provide the following operators: +, -, /, *, <, >, <>, And, Or, Not, If.

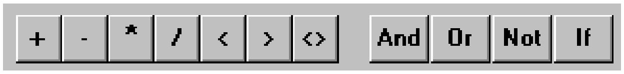


Figure B-3. Syntax Buttons in the Expression Builder window

"If" results in IIf(<Expression>,<TruePart>,<FalsePart>), see also Section B.4, If-then-else Statements.

You can also use other operators, but only these, the most commonly used operators, have their own buttons in the window.

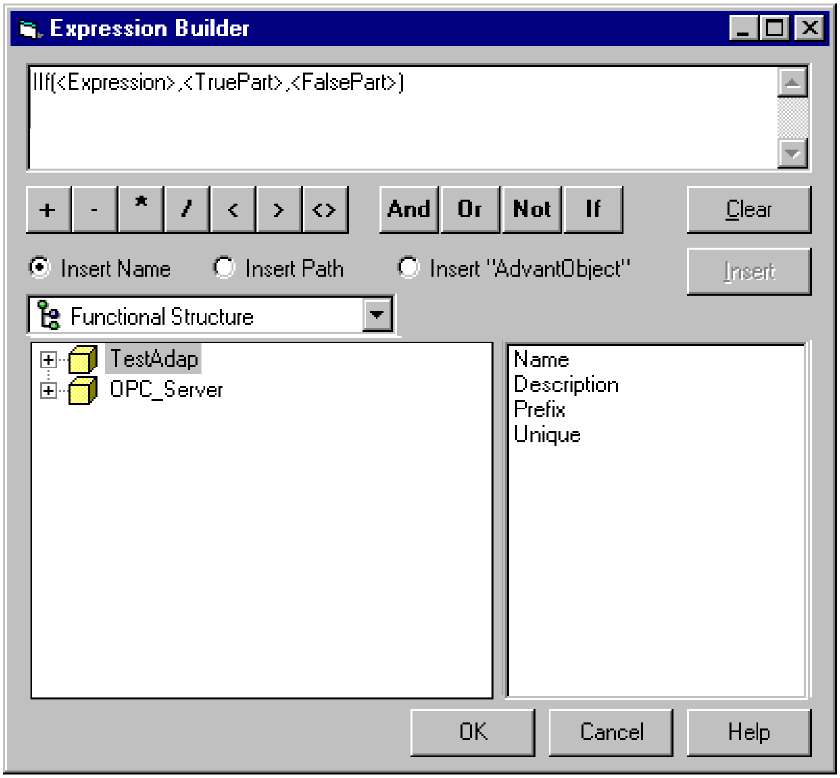
For more information about Visual Basic operators, see Microsofts On-line Help and On-line Documentation for Visual Basic.

B.4 If-then-else Statements

IIf in Visual Basic allows you to build an if-then-else statement in the expression style. The Expression Builder window give you some help with the syntax of the if-then-else statement.

When you click on the **If** button, the following "template syntax" is presented in the Edit box:

IIf(<Expression>,<TruePart>,<FalsePart>)



IIf(<Expression>,<TruePart>,<FalsePart>)

● Insert Name ○ Insert Path ○ Insert "AdvantObject"

Functional Structure

TestAdap
OPC_Server

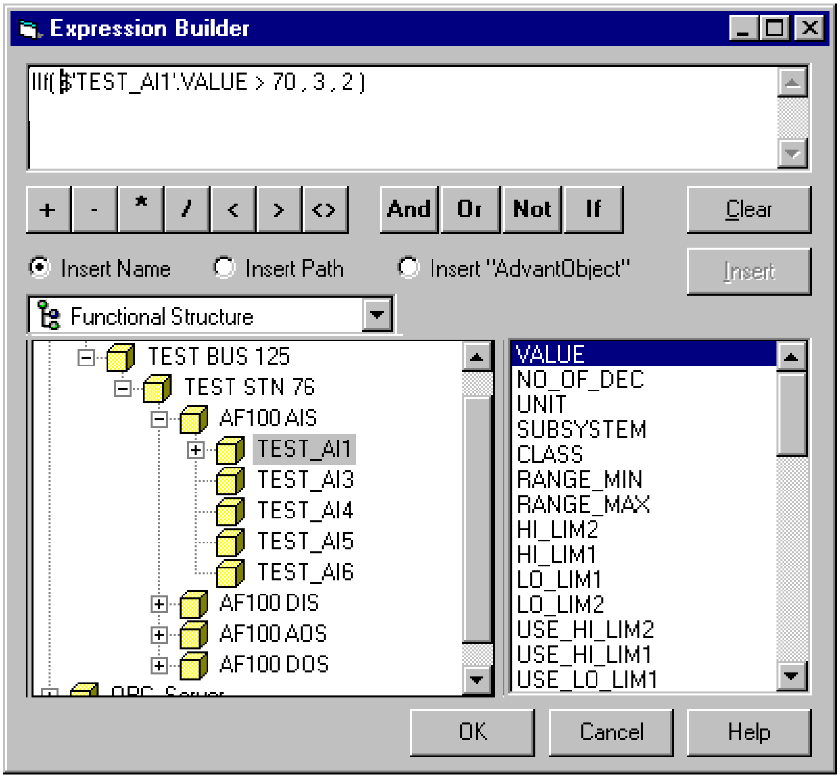
Name
Description
Prefix
Unique

Figure 2-4. Expression Builder window with a "template syntax"

You can then select and replace the different portions. In most cases, the inserted symbol gets a blank space on both sides.

We can for example replace the "template syntax" with the following expression:

IIf($'TEST_AI1'.VALUE > 70, 3 , 2)



IIf($'TEST_AI1'.VALUE > 70 , 3 , 2)

● Insert Name ○ Insert Path ○ Insert "AdvantObject"

Functional Structure

Figure 2-5. Expression Builder window with an if-then-else statement

which means that the value "3" is presented if the VALUE property of the object TEST_AI1 has a greater value than 70. If not - the value "2" it will presented.

When no value is available, the value field is set to NULL. The significance of that is that expressions that deliver a boolean result always evaluate to false when one or more of the input variables has the value NULL. (Except if the expression explicitly checks for the NULL value). This can be used when the if-then-else expression style is used (when iif is used). It is always the last “then” clause that is used when data has not been delivered. In the following example the expression evaluates to red when data has not been delivered yet:

```
IIf($’AI.117’.value < $’AI.117’.HiLim, green, red)
```

B.5 Paths to the OPC Server

If more than one ABB Object have the same name, the expression syntax must be able to identify the correct OPC items in the OPC server. This is solved by adding the path to the input property in the expression.

If we select **Insert Path** and insert an OPC item in an expression we automatically add the path for the selected OPC item.

For example, without Insert Path selected:

```
$’TEST_AI1’.VALUE *40
```

and with Insert Path selected:

```
$’TestAdap/TEST BUS 125/TEST STN 76/AF100 AIS/
TEST_AI1’.VALUE *40
```

which means that the TEST_AI1 property VALUE (located at TestAdap/TEST BUS 125/TEST STN 76/AF100 AIS) is multiplied by 40.

INDEX

Numerics

3D effect 149
3D frame properties 149

A

ActiveX
  - ActiveX components 34
  - add external ActiveX controls 9, 34

ActiveX components 10
ActiveX controls
  - installing external (third-party) ActiveX controls 9

Add Member dialog box 39
Add Member window 38
Adding
  - members 37, 112
  - methods and events 41
  - properties 42

Advanced User mode 23
Advant Extended Primitives 33
Advant Graphics
  - product overview 4

Advant Graphics Builder
  - Advant Graphics wizard 16
  - application start-up 14, 81
  - optional preferences 22
  - specific features 13, 80
  - user interface 13, 80
  - window 15, 82
  - wizard 98
  - working with 14, 82

Advant Graphics Studio
  - what you can do with 4

Advant Options 22
Advant options
  - Advanced user mode 23
  - Debug 23
  - Live data in design mode 22

Advant Plant Explorer 96
Advant Primitives 33 to 34, 111
  - AdvantArc 121
  - AdvantBar 122
  - AdvantBezier 123
  - AdvantChord 124
  - AdvantCone 125
  - AdvantEllipse 126
  - AdvantHscale 127
  - AdvantLimit 128
  - AdvantNumeric 129
  - AdvantObjecttrim 131
  - AdvantPie 133
  - AdvantPipe 134
  - AdvantPolygon 135
  - AdvantPolyline 136
  - AdvantRectangle 137
  - AdvantText 138
  - AdvantVscale 139
  - Aspect Link 141

Advant Subelements 33, 99
  - AdvantBar 153
  - AdvantObjectTrim 159
  - AdvantStatusLine 160
  - AdvantValueField 158

AF100 Test OPC Server (Off-line data source) 29, 86
Angle properties 151
Application start-up
  - accessing process dialogs 62
  - create a new aspect 14, 81
  - edit an existing aspect 14, 81

Auto load
  - see Process dialog window
    - automatic configuration (auto load) 71

B

Basic building blocks provided by ABB
  Advant Primitives 33, 121, 153
  Advant Subelements 33
Bitmap picture
  scaling 87
Bounding rectangle 30
Building blocks provided by ABB 119
Building expressions 49

C

Changing the process dialog user profile 60
Configuration View 68
Configuring
  the process dialog 60
Copying
  graphic aspects in the same object structure 56

D

Debug 23
Design Mode 105
Design mode
  Live data in Design mode 22

E

Element Browser window 100
Equal button 50, 103
Expression Builder window 51, 103, 166
  syntax 170
Expressions 165
  ABB expressions 166
  assigning expressions 46
  assigning expressions to properties 47
  building expressions directly in the Expressions window 49, 101
  building expressions in the Expression Builder window 49, 103
Expressions window 47, 102, 104, 166
External ActiveX Controls 34

F

FAQ 93
Fill properties 143 to 144, 146
Fill style property 145, 147
Frequently asked questions (FAQ) 93

G

Graphic displays 79
  building 83
  deleting 87
  deploying 87
  editing 84
  saving 87
  testing 86
Graphic Element Browser 21, 45
Graphic Element Browser window 45
Graphic elements 11, 120
  adding graphic elements to a graphic 44
  assigning an expression to a property 102
  building 23, 99
  changing the properties 101
  deleting 30
  deploying 30
  editing 27
  saving 30
  searching 44
  testing 29
Graphics
  viewing 89

I

If-then-else statement 170
Importing Visual Basic projects 57
Insert Path 172
Installation 9
Installing External (Third-party) ActiveX Controls 9

L

Line style property 143

Live Data in design mode 22

M

Main Window 63
Maintenance 93
Member types 40

O

Object Display 66
Object references 167
Off-line data source (AF100 Test OPC Server) 29, 86
OPC items 51, 53, 166, 172

P

Point properties 151
  to add a new point 151
  to delete a point 152
  to move a point 152
Preparations
  installation 9
Primitive elements 11, 120 to 121
Primitive elements provided by ABB
  3D effect 149
  3D frame properties 149
  angle properties 150
  fill properties 144
  fill properties with shade 146
  fill styles 145, 147
  line properties 143
  line styles 143
  point properties 151
  smooth 150
Process dialog window 62
  automatic configuration (auto load) 71
  configuration view 68
  main window 63
  manual configration 70
  object display 66
Process dialogs 59
  changing the process dialog user profile 60, 77
  configuring 60, 72
  process dialog profile definition 75
  special features 61
  user interface 61
  views 61
  working with process dialogs 62
  working with the user profile 73
Product verification 10
Properties
  for primitive elements provided by ABB 142
Properties window 42, 101, 167
  Scaling bitmap pictures 88

R

Relative references 54
  display property ObjectName 102, 116
  relative name AdvantObject 54, 102, 104
  when copying graphic aspects 56

S

Scaling
  Bitmap pictures 87
Searching for graphic elements 44
Special characters 44
Subelements 12, 120, 153
Syntax buttons 52, 169
Syntax for item references 168
  ObjectRef part 168
  PropertyRef 168

T

Template syntax 170
Test Mode 104
Toolbox
  adding tools 35
  deleting tools 37
  providing elements from the toolbox 35
  using 100
  working with the toolbox 30

Tools window 36
Tutorial 95
  assigning expressions to the graphic element 101
  building a graphic element 99
  building expressions for the graphic element 103
  changing properties for thegraphic element 101
  creating a graphic element and using Visual Basic code to animate it 111
  creating a graphic element from Advant Plant Explorer 96
  deploying a graphic element 105
  saving a graphic element 105
  testing the graphic element 104
  using the graphic element in a graphic display 105

V

Visual Basic operators 169